DOCUMENT RESUME

ED 263 884 IR 011 870

AUTHOR	Van Dusseldorp, Ralph, Ed.
TITLE	Education=Microcomputers, Classrooms, and Children. Proceedings of the Annual Statewide Conference of Alaska Association for Computers in Education (4th, Anchorage, Alaska, April 4-6, 1985).
INSTITUTION	Alaska Association for Computers in Education, Anchorage.
PUB DATE	Apr 85
NOTE	195p.
AVAILABLE FROM	Alaska Association for Computers in Education, P.O. Box 4-652, Anchorage, AK 99502 ($10.00).
PUB TYPE	Collected Works - Conference Proceedings (021) -- Guides - Non-Classroom Use (055) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC08 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; Computer Literacy; Computer Software; Conference Proceedings; Elementary Secondary Education; Instructional Material Evaluation; *Microcomputers; *Programing; Programing Languages
IDENTIFIERS	*Alaska; *Computer Uses in Education

ABSTRACT

These proceedings include 31 papers presented at the 1985 conference of the Alaska Association for Computers in Education. Generally oriented toward the use of microcomputers in elementary and secondary schools, papers deal with both administrative and instructional microcomputer applications. Topics addressed include: microcomputer literacy models, distance education, trends and issues in Logo, computer assisted curriculum development, software copyrights, integrating computer software into the regular school curriculum, PLATO and EDNET, developing courseware to meet the needs of native Alaskans, writing with Quill, Apple Logo projects, use of a microcomputer controlled videodisc player with mentally handicapped students, and sex differences in computer involvement. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

BEST COPY AVAILABLE

PROCEEDINGS

OF THE
FOURTH ANNUAL STATEWIDE CONFERENCE OF

ALASKA ASSOCIATION FOR COMPUTERS IN EDUCATION

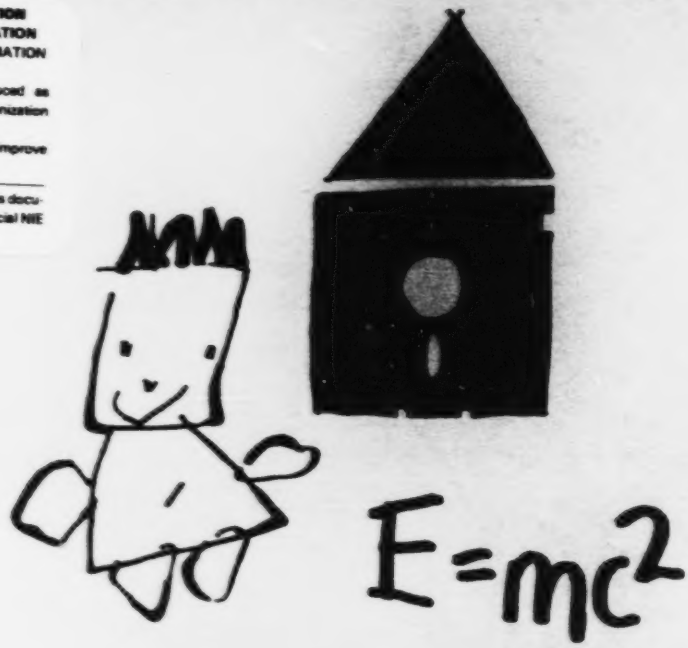
ED263884

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.



EDUCATION
=
MICROCOMPUTERS, CLASSROOMS,
and
CHILDREN

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Ralph Van Dusseldorp

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

CAPTAIN COOK HOTEL • ANCHORAGE • APRIL 4-6, 1985

IR011 870

PREFACE

This is the second time we have published AACED conference proceedings. We hope the Proceedings will be helpful to persons attending the conference in choosing sessions to attend, gaining information about sessions they are not able to attend, taking notes during sessions, and serving as a reference later. It is also hoped that the Proceedings will serve as a valuable source of information for those who were not able to attend the conference.

After the conference we will evaluate the Proceedings and will decide whether to make changes in the future. We appreciate receiving your comments concerning the Proceedings.

Special appreciation is expressed to the authors of the papers and presentations that are included in this document. They all submitted them well ahead of the conference so this document could be available for distribution at the conference.

Appreciation is also expressed to Mike Chmielowski, Conference Program Chairperson, who helped communicate with authors, and Jim Bryan who assisted with the layout and arrangements for printing.

Ralph Van Dusseldorp
Editor

CONTENTS

Page

Helen Barrett
Micro Computer Literacy Competency Models 1

Carole Bartholomew
Selecting and Evaluating Software for the Adult Learner 13

Anne Batey
Make Them Think! Generating and Using Classroom Data Bases 14

Sandra Berry
State Level Curriculum: The Power, Influence and Opportunity 22

William Bramble
Distance Education 23

C. M. Chmielowski
Keyboarding: Helping Students Learn About Learning 28

Scott Davies
Speech Synthesis for Early Elementary 40

Mike Dishnow and Bonnie Murrill
Computer Scheduling 46

Mary Alice Felleisen
First Level Mathematics (KINDERMATH II) 49

Thomas Gaylord
The Use of Program Evaluation and Review Techniques (PERT)
in Educational Administration 51

Gary Gustafson
Turbo Pascal in the High School Computer Lab 55

Mary Claire Harris
Library Media Center Management with Microcomputers 63

Mary Susan Henrikson
Trends and Issues in Logo: Implications for Teachers 70

Michael Hoar and Ken Klawunder
Computer Assisted Curriculum Development and Resource Management 77

Daniel Hoffman
Dialog Workshop; Using a Dialup Database in the Classroom 79

Page

Norman Holthouse
Copyright and Copying; Rights and Responsibilities 85

Norman Holthouse and Ralph Van Dusseldorp
The Results of an Experiment with a Computer in Each Classroom 90

Mike Johnson
Spread Sheet and Data Base Software 98

Ray Lehrman, Will Files, Jan Ingram and Patt Gibbs
PLATO and EDNET 121

Robert McCabe
Postsecondary Training Options Through the Alaska Department of Education 122

Alan McCurry and Niki McCurry
If the Shoe Fits They'll Wear It: Project APEL, A Holistic process Oriented Approach Enabling Every Teacer at every school to integrate the use of Computers into the regular school curriculum K-12 123

Niki McCurry and Alan McCurry
Alaskan Software for Alaskan Students: Developing Coursework to meet the needs of Native Alaskans 125

Louise Petermann
Matching Computer Software to the Elementary School Curriculum 127

Don Putnam
The Use of Commercially Available Utility Programs by the Student Programmer 130

Ethna Reid
Basic Literacy Through Microcomputers (BLTM): Teaching Computer Keyboarding, Reading, Spelling and Typing Skills 133

Ron Richter and R.W. McFarland
CP/M and More, For Your Apple 139

Marcia Romick
Writing with Quill 142

Donna Gail Shaw
Apple Logo Projects for Teachers and Children 145

Nancy Parks Sopp
What Should Children be Learning with Computers 164

Page

Ron Thorkildsen
Using a Microcomputer Controlled Videodisc Player in the Education of Mentally Handicapped Students 170

Ralph Van Dusseldorp and Norman Holthouse
Differences Between Male and Female Involvement with Computers in Schools 180

MICRO COMPUTER LITERACY COMPETENCY MODELS

A Tool for Staff Development

Developed by

Helen Barrett

Fairbanks North Star Borough School District
Community Computer Lab Facilitator
Staff Development Micro Computer Trainer

ABSTRACT

Current literature in adult learning, training and development, advocates the use of Competency Models as a tool to facilitate the learning process, and as a guideline for training and for job competence assessment. During the last year, a Micro Computer Literacy Competency Model has been under development in the Fairbanks North Star Borough School District. This model includes lists of skills needed at various occupational levels within the district (teacher/professional, manager/administrator and clerical/support staff). At a different level, a list of very specific skills has been developed for using functional software tools, such as Word Processing, Data Base Management, Electronic Spreadsheets, Telecommunications, Graphics, Hardware and Operating Systems.

USING COMPETENCY MODELS AS A LEARNING TOOL

One of the major problems faced by learners in any discipline is knowing the scope of the field and the particular components that make up a body of knowledge and skill. Finding the answers to questions about a particular area

is not possible if a learner doesn't know "what questions to ask" at the beginning of a learning experience. As we progress in learning a field, knowing what is expected to be considered "competent" in that field provides a model from which we can evaluate our current level of knowledge and skill, the level of competency required, and the "gap" that determines a **learning need**. This learning "gap" is illustrated below:

REQUIRED LEVEL OF COMPETENCY ⬅ LEARNING NEED ➡ PRESENT LEVEL OF COMPETENCY

> The more concretely individuals can identify their aspirations and assess their present level of competencies in relation to them, the more exactly they can define their educational needs--the more intensely will they be motivated to learn. And the more congruent the needs of individuals are with the aspirations their organizations and society have for them (or the other way around--the more congruent the aspirations of organizations and society are with the educational needs of individuals), the more likely will effective learning take place. (Knowles, 1980, p.88)

This is the purpose of a **Competency Model**: to provide a complete definition of the Knowledge and Skill requirements in areas relevant to the performance of specific types of work. A competency model outlines what you have to know (knowledge) and be able to do (skill) to be effective in a particular job or discipline. As further definition, a ***competency*** is "any attribute of a person that underlies effective performance; a job competency is simply an attribute related to doing a job effectively" (Klemp, 1979).

> A personal attribute in this context can take many forms--it can be a kind of knowledge, an ability, an interest, a trait, a motive, even a self-concept--but the only way that such an attribute becomes a competency is in its relationship to performance. People carry with them a wide assortment of knowledge, abilities, interests, traits and motives, but unless these attributes relate demonstrably to doing a job well, they are not job competencies. (Klemp, 1982)

There has been a lot of attention paid recently to **Computer Literacy**, as a current learning need, and job competency for many people in our society.

There are, however, no clear guidelines as to what it means to be competent in the area of Computer Literacy. This document is an attempt to define some of these competencies, as a guide for learners as they begin their own efforts to become competent users of micro computer technology. Once developed, a competency model can serve as a guideline for providing staff development training, as well as for individuals developing an Individual Computer Literacy Education Plan (ICLEP, as advocated by ICCE's Dave Moursund). Staff development trainers can use the model to insure uniformity of material covered in structured classes. Independent learners can use the model to develop learning contracts and to use as a checklist of specific skills mastered.

DEVELOPING A COMPETENCY MODEL

There have been several articles published recently on the development of Competency Models in general, including several that describe the process specific to the use of computers in education. The November, 1984, issue of The Computing Teacher has two articles that specifically address the subject of Computer Competencies for Educators. William E. Baird's article, "A Road Atlas for Computer Literacy and Teacher Training," presents a taxonomy of **(1) concerns, (2) need for skills** and **(3) level of proficiency** in an educational setting. This article emphasized the need to provide a clear **working definition** of computer literacy as a **process** (not an event) which "is made up of a graduated sequence of skills which grow out of a similar sequence of needs." Baird also provides valuable suggestions for implementing staff development training to move up his competency grid, illustrated as a three-dimensional block, with each side representing these

levels of concerns, need for skills and proficiencies (see Appendix A). In the same issue, in an article entitled, "Microcomputer Competencies for Vocational Teachers," Gene L. Roth and Dennis G. Tesolowski described the process they used to develop a similar competency model with vocational education teachers in Illinois. This process, similar to the method used to develop competency models with adult educators throughout Alaska by the late John Hulbert, resulted in a profile that consisted of 47 competencies clustered within five categories:

A. Developing a personal plan for microcomputer competency
B. Integrating computer-based instruction (CBI) into vocational curricula.
C. Planning, executing and evaluating CBI
D. Planning and organizing vocational education learning environments for CBI.
E. Performing classroom management functions with CBI.

In 1983, the American Society for Training and Development (ASTD) completed a competency study entitled Models for Excellence, a *prescriptive* model for the training and development field. The study team was charged with the task "to produce a detailed and updatable definition of excellence in the training and development field in a form that will be useful to and used as a standard of professional performance and development by ASTD, organizations, educational institutions, training and development departments, and individuals practicing or expected to practice in the training and development field." The ASTD Competency Study Methodology consisted of the following steps:

Determine the domain of the T & D field
Determine the key roles in the T & D field
Identify the major environmental forces expected to affect the field in the near future.
Identify the critical outputs of the T & D function.
Identify the critical competencies for the T & D field.
Develop behavioral anchors for the competencies
Cluster the roles to reflect common competency requirements.

These steps used by ASTD, plus the processes described in The Computing Teacher, provide the conceptual framework for the development of a Micro Computer Literacy Competency Model used in Staff Development Training in the Fairbanks North Star Borough School District. The steps being implemented in the FNSBSD include:

1. Development of a K-12 Computer Curriculum.
2. Definition of computer literacy skills
   A. Levels of Sophistication in use of computers by role/occupation
   B. Specific skills needed for various occupations
3. Functional software skills, needed in a variety of occupations.

DEFINITION OF MICRO COMPUTER LITERACY

There are a variety of definitions of computer literacy. In the winter of 1982-83, the Alaska State Department of Education conducted a Statewide Computer Literacy Study, in which all current definitions were reviewed and synthesized. The workgroup on citizen computer literacy recommended the following definition of computer literacy:

> Computer Literacy for the State of Alaska means developing all of our human resources in an information and communications-based society. A society that fosters computer literacy is one that affords its citizens opportunities to acquire computer literacy skills, knowledge and attitudes appropriate to individual needs. (AKDOE, 1982, P.v)

This definition is not specific as to what these individual needs are and gives very little direction for determining specific areas of study. Other definitions are equally vague:

1. Computer Literacy is whatever a person needs to know and do with computers in order to function competently in our information-based society. (Hunter)
2. Computer literacy is a functional knowledge of computers and their effects on students and on the rest of society. This knowledge should be at a level compatible with other knowledge and skills a student is acquiring in school. (Moursund)

MICRO COMPUTER LITERACY COMPETENCY MODELS (Continued)

3. There are six skills essential to becoming computer literate:
   How to operate computers;
   How to use computer applications;
   How to integrate applications into [your work]:
   How to evaluate applications;
   How to design new applications;
   How to program computers. (Rawitsch)

4. If you can tell a computer what you want it to, you are computer literate; if you can't, you're not. Computer literacy is the ability to do computing. (Leuhrmann)

5. Computer literacy is whatever understanding, skills and attitudes one needs to function effectively within a given social role that directly or indirectly involves computers. (Anderson and Klassen)

None of these definitions provides enough detail for individuals learners to make intelligent decisions about specific learning activities. These definitions also do not separate the skills needed to effectively use a micro or personal computer, as opposed to larger mini- or main-frame computer systems. Since a vast majority of people in our society will use the smaller **micro** computers, this document is limited to developing a model that fits that level. To provide further manageable limits for this study, it is necessary to define several levels of competencies:

A. **Level of sophistication in Social Roles or Occupations**

B. **Specific skills for various occupations**

C. **Functional skills in Use of Applications Software.**
   A detailed inventory of skills has been developed in:
   - Micro Computer Hardware and Operating Systems
   - Word Processing
   - Data Base Management
   - Electronic Spreadsheets
   - Telecommunications
   - Graphics (Converting numerical data into graphics)

MICRO COMPUTER LITERACY COMPETENCY MODELS (Continued)

A. **Level of sophistication in Social Roles or Occupations**
(Provided by State of Alaska's Statewide Computer Literacy Study)

Levels	Descriptions
1. Computer Awareness	Needed by all citizens (e.g., history, concepts, hands-on experience)
2. Entry Level Use	Use computer applications designed by other people (e.g., data entry clerk, etc.)
3. Computer Use	Use computer as available tool to do job (e.g. spreadsheet, accounting, data base management, word processing)
4. Computer Creation	Employ computer as tool to create a tool (e.g., engineering, architecture, modeling)
5. Data Processing	Help other people do their job with computer (e.g., systems analyst, programmer)
6. Computer Systems Development	Design hardware and software for computer systems (e.g., software vendor, systems engineer, systems programmer)

Most staff development micro computer literacy training in the FNSBSD has been focused at Levels 1 through 3 above (Awareness, Entry Level Use and Computer Use of software tools). This training has been facilitated through credit classes provided through the University of Alaska, Tanana Valley Community College and non-credit classes through the FNSBSD Community Schools program. In addition, in the fall of 1984, a special program of individualized training and technical assistance was developed to facilitate the use of the IBM PC XT micro computers recently purchased for use in the district's Central Office. This training included formal classes in various software applications, integrated with desk-level technical assistance.

MICRO COMPUTER LITERACY COMPETENCY MODELS (Continued)

B. EXAMPLES OF SPECIFIC SKILLS NEEDED IN VARIOUS OCCUPATIONAL CATEGORIES:

TEACHER/PROFESSIONAL: (Level 3)

Minimum Competencies * PLUS:

Teaching Problem-Solving with the Computer (programming in LOGO or BASIC)

Understand applications for various types of Computer-Assisted Instruction: Drill & Practice, Tutorial, Simulations, Modeling

Functional skills using software tools, such as Word Processing, Data Base, Spreadsheets, Telecommunications, Graphics, Operating Systems

Members of the Computer Curriculum Committee of the FNSBSD are in the process of defining a competency model/training guide for Teacher/Professionals in the district. A copy of the first draft of this model will be available at the conference. This visual "roadmap" starts at the center, with minimum computer skills, and branches out into operational areas of interest to each teacher, depending on individual needs and/or curriculum areas. Each of these areas further defines functional software skills, for which a fairly complete inventory of skills has been developed.

ADMINISTRATOR/MANAGER: (Level 3)

Minimum Competencies * PLUS:

Be able to use a micro computer as a tool within the functions of management: Planning, Organizing, Controlling

Functional skills using Software Tools, such as Data Base, Spreadsheets, Telecommunications, Graphics, Operating Systems

The model for Administrator/Managers will be developed over the next year, as more administrators begin using micro computers.

CLERICAL/SUPPORT STAFF: (Level 2 & 3)

Minimum Competencies * PLUS:

Functional skills using Software Tools, such as Word Processing, Data Base, Spreadsheets, Telecommunications, Operating Systems

The model for Clerical/Support Staff is being developed with and for secretaries in the FNSBSD Central Office. This work will be further expanded as the new Student Information System is implemented and will place micro computers in every school office over the next two years.

* A model for minimum computer literacy competencies has been published by Roger L. Sanders in "Computer Literacy: Innovation Adoption and In-service Training" from Microcomputers for Adult Learning, edited by David G. Gueulette. Follett Publishing Co. (1982). See Appendix B.

MICRO COMPUTER LITERACY COMPETENCY MODELS (Continued)

C. FUNCTIONAL SKILLS IN USE OF APPLICATIONS SOFTWARE

In the interest of brevity, this list has been shortened to the major topics. The complete list will be available during the conference presentation. Information for this part of the document was adapted from course outlines in IBM's Guide for Learning.

1. **Hardware & Operating Systems**
   - A. Familiarity with micro computer terminology and learning resources (manuals, etc.)
   - B. Familiarity with operation of hardware system
   - C. Ability to perform disk housekeeping with Disk Operating System Commands
   - D. Practical Troubleshooting.
2. **Word Processing Skills**
   - A. Entering Editor, Typing text, Printing text
   - B. Moving cursor and performing simple editing
   - C. Inserting and Deleting Lines and Text Segments
   - D. Changing Margins, Line Spacing and Tabs.
   - E. Moving, Deleting, Reading and Writing Blocks of Text.
   - F. Integrating various operations
   - G. Performing Global Searches and Replacements
   - H. Changing Page Formatting.
   - I. Embedding Printing Commands
   - J. Deleting, Copying and Renaming files.
   - K. Using Spelling Checking programs
   - L. Developing form letters, Merging data files into text, Addressing envelopes.
   - M. Preparing files for transmission via modem.
   - N. Using special features of some software, such as glossary, columns, math, link and group text.
   - O. Integrating word processing program with various printers.
3. **Data Base Management Programs**
   - A. Getting acquainted with the Program
   - B. Searching for Data in a File
   - C. Adding Data to a File.
   - D. Sorting and Printing
   - E. Designing and Redesigning a File
   - F. Managing Files and Disks
4. **Electronic Spreadsheets**
   - A. Orientation to Spreadsheets
   - B. Building a Spreadsheet: Primary Commands and Functions
   - C. Replication/Copying
   - D. Modifying and Displaying the Worksheet
   - E. Advanced Functions and Commands
   - F. Printing and Documentation
   - G. Advanced Topics/ Template Creation and Project Development
5. **Telecommunications Skills**
   - A. Communications Function: Hardware and Software Needed
   - B. Communicating with another computer.
   - C. Gain Access to Packet-Switching networks
   - D. Using Electronic Message Systems
   - E. Using Information Utilities and Retrieval Services
6. **Graphics**
   - A. Constructing Rectangular Coordinate Systems
   - B. Constructing Bar Graphs, Line Graphs, Pie Graphs, Function Graphs
   - C. Constructing Graphs from DIF Files

MICRO COMPUTER LITERACY COMPETENCY MODELS (Continued)

APPENDIX A - TAXONOMY OF LEVEL OF CONCERN, NEED FOR SKILL AND PROFICIENCY - William E. Baird.

Level of Concern

0. **Awareness** - Lack of concern, perhaps resistance to change
1. **Informational** - Desire to know more about the process.
2. **Personal** - Concern for the impact of the process in the individual.
3. **Management** - Concern for finding time and resources to learn about the process.
4. **Consequence** - Thoughts of how the process will impact one's students.
5. **Collaboration** - Desire to relate new experiences with the process to others.
6. **Refocusing** - Sufficient experience with the process to create variations and improvement.

Level of Need for Skills

0. **Pre-Skill** - Entry level; may be accompanied by some anxiety about computers.
1. **Orientation** - Ability to turn on the computer, load a program from tape or diskette and use prepackaged software by following directions from the screen.
2. **Evaluation** - Ability to appropriately use and critically evaluate most available software for the user's grade level; can select from a variety of software that which is best suited for the classroom.
3. **Application** - Ability to use the computer as a learning "amplifier" in the classroom; drill and practice, computer-assisted instruction, tutorials, games and simulations are included in the repertoire of this user.
4. **Programming in the Small** - Ability to read, write and modify short computer programs of less than 300 lines in a single language such as BASIC or Logo; may teach simple programming.
5. **Programming in the Large** - Ability to write larger programs in at least two high-level languages using a procedural approach so that the problem to be solved by the program is broken up into smaller sub-tasks, each of which is a subroutine in the larger program.
6. **Computer Science Concepts** - Ability to apply sound principles of computer science to the selection of appropriate data structures and algorithms in routine authoring of programs longer than 600 lines.

Level of Proficiency

0. **Discovery** - First encounters with the new skill. Here the teacher must decide if the new skill is potentially useful. Until the skill is valued, no further skill acquisition is indicated.
1. **Skill Building** - The learner practices using the new tool that was learned through discovery. First practice should be in the introductory session, but only through use in the classroom with the skill be mastered.
2. **Design** - Occurs after mastery of the new skill, and after enough practice to see ways of varying or improving the tool in a local environment. This could be followed by modifying the tool or creating a parallel tool.

Source: Baird, William E. "A Road Atlas for Computer Literacy and Teacher Training." The Computing Teacher, November, 1984, pp. 11-16

APPENDIX B. MINIMUM MICRO COMPUTER LITERACY COMPETENCIES

MICROCOMPUTER HARDWARE

1. Able to identify the components of a microcomputer system.
2. Demonstrate an understanding of the function of these components.
3. Able to operate a microcomputer in relation to its use on the job. **
   **(See section on various occupations for specific applications.)

MICROCOMPUTER SOFTWARE *

1. Able to identify the types of computer software
2. Able to identify evaluative criteria used in assessing software quality.
3. Aware of the software available within the organization.
   *(See section on software tools, such as Word Processing, Data Base Management, Electronic Spreadsheets, Telecommunications, Graphics and Operating Systems

MICROCOMPUTER PROGRAMMING

1. Aware of the need for using a language understood by the computer and able to identify one computer language.
2. Able to write a simple computer program.

APPLICATIONS OF COMPUTERS

1. Aware of the diverse use of computers in everyday life.
2. Aware of the applications of the microcomputer as a learning, working and problem-solving tool.

IMPACT OF COMPUTERS ON SOCIETY

1. Aware of the effect of applying computer technology in society.
2. Able to identify positive and negative results of the impact of computers.

INDIVIDUAL ATTITUDES TOWARD COMPUTERS

1. Demonstrate attitudes that reflect a lack of fear, anxiety, or intimidation related to computers.
2. Value increased communication and the availability of information through computers.
3. Feel confident about the ability to operate a microcomputer.

These objectives are aimed at developing minimum computer literacy competencies. ** Participants should be able to demonstrate mastery of these objectives. The computer literacy competencies outlined are designed to increase general awareness levels about computers and to develop specific operational skills needed to use microcomputers in a variety of applications.

** Source: Sanders, Roger L. "Computer Literacy: Innovation Adoption and In-service Training" from Microcomputers for Adult Learning, edited by David G. Gueulette. Follett Publishing Co. (1982)

BIBLIOGRAPHY

Alaska Department of Education, Office of Educational Technology & Telecommunications. Statewide Computer Literacy Study. Prepared by: Systems for Training and Applied Research, Inc. 1983.

Baird, William E. "A Road Atlas for Computer Literacy and Teacher Training." The Computing Teacher, November, 1984.

Hunter, Beverly, et. al. Guide to Learning. Internation Business Machines Corporation, 1984

Ingalls, John D. Managerial Success: A Guide for Improving Performance and Developing Competence. Arlington, MA: Competency Development Corporation, 1980.

Klemp, George O., Jr. "Job Competence Assessment: Defining the Attributes of the Top Performer." Reprint for ASTD Research Series, Vol. No. 8., 1982.

Klemp, George O., Jr. "Defining and measuring competence." In P. Pottinger & J. Goldsmith (Eds.), Defining and measuring competence. San Francisco: Jossey-Bass, 1979.

Knowles, Malcolm S. The Modern Practice of Adult Education. Chicago: Follett Publishing Co., 1980

McLagan, Patricia A. Models for Excellence. American Society for Training and Development, 1983.

Moursund, David. "ICLEP (Individual Computer Literacy Education Plan): A Powerful Idea" The Computing Teacher. November, 1983.

Roth, Gene L., and Tesolowski, Dennis G. "Microcomputer Competencies for Vocational Teachers." The Computing Teacher, November, 1984.

Sanders, Roger L. "Computer Literacy: Innovation Adoption and In-service Training" from Microcomputers for Adult Learning, edited by David G. Gueulette. Chicago: Follett Publishing Co, 1982.

SELECTING AND EVALUATING SOFTWARE FOR THE ADULT LEARNER

Carole Bartholomew

Selection and evaluation of software for the adult learner is a critical process. What may be perceived as "good" software at the elementary or high school level may not be appropriate for the adult learner for several reasons: readability, graphics, language, and audience. It is important, therefore, to ascertain reasons and methods for selecting and evaluating software for the adult learner. Another concern is that sources of computer software for postsecondary work are limited. In a recent article on computers in higher education, fewer than a dozen publishers of college-level instruction were listed. After locating software, evaluation must take place regarding the appropriateness of the program. Software reviews and evaluations can be found through commercial sources (such as EPIE, microSIFT, RICE), professional journals and groups (such as AEDS, National Council of Teachers of Mathematics), local/regional organizations (such as the Region X Adult Education Software Consortium, Alaska Association of Computers in Education, Northwest Council of Computers in Education), and peer and professionals in the field.

MAKE THEM THINK! GENERATING AND USING CLASSROOM DATA BASES

Anne E. Batey
Computer Education Specialist
Northwest Regional Educational Laboratory
Portland, Oregon

"Knowledge is Power." Edward A. Figenbaum (and others)

"There is no automatic link between Knowledge and Power."
Langdon Winner

INTRODUCTION: The Case for Classroom Data Base Use

The information age is here. Students of today are being raised in the information age. As information acquisition becomes easy and rapid, what happens to it? What good is information if students can't do something USEFUL with it? What is the link between Knowledge and Power? A critical skill for students of today is the ability to manipulate and process information. Analyzing, synthesizing, and evaluating--those familiar, higher-level cognitive skills, will be more important than ever in an age where information is plentiful.

> Problem: As the Information Age increasingly demands thinkers who can make information useful, How do we prepare such "information processors"? How do we provide opportunities to develop and practice "information processing" skills?
>
> Solution: The computer, the vehicle for the information acquisition, is also the vehicle for practicing information processing.

In the "old days" students went to the library, read encyclopedias, reworded the sentences they read, and wrote reports. The "research" process was largely information acquisition and regurgitation. The skilled report writer was able to find more sources and "reword" the best. In the age of electronic information acquisition, the data base

regurgitates, someone else--the student, must process it. A data base of specific information allows a student to search, sort, and select information very rapidly. A student can explore and manipulate "raw data" in a way not possible prior to computers. A data base is ideal for arranging information to see patterns and relationships. The computer takes care of the tedious part and does it very fast; the student is left with the challenge of making sense of the information. As a teacher this is exciting; in making sense of information, students will need to compare, interpret, analyze, synthesize, and evaluate. In doing so, they are learning how to make information useful and ultimately, make knowledge powerful.

AN EXAMPLE: Data Base Use in the Chemistry Classroom

The following description presents a classroom activity in two scenarios. The first took place prior to computers and data bases. The second scenario describes the same classroom activity conducted with computers and data base software. I have included this activity to model how an existing activity can be enhanced by data base use.

Scenario BC (before computers):

Every spring following a semester of general inorganic chemistry and a unit on organic chemistry, I incorporated an Issues-in-Science unit on pesticides. Students were asked to "Adopt-a-Pesticide", go to the library, use the chemical research materials, write a report on their pesticide and make brief oral presentations on their findings.

The first week of the unit was devoted to information acquisition. The students had 7-10 categories on which they were to find information

(see Appendix A). During the second week the class listened to the oral reports and we discussed the pros and cons of using a particular pesticide. For example: We would contrast a highly effective but persistent and toxic herbicide VS a herbicide that was more expensive, less effective but more likely to degrade into harmless substances. The activity was worthwhile but not a smashing success.

The strengths of the activity:

1) Students learned to use some of the chemical reference literature to find chemical formulas, to find information on commercial manufacturers, etc.

2) Students observed that their knowledge of chemistry was sufficient to use and understand the information sources that the "real scientists" use.

3) The students enjoyed the break in routine.

The weaknesses of the activity:

1) The activity did not lead students to form generalizations about all pesticides. Typically, I had to follow the oral reports with more information. For example: I would point out that the chlorinated phenyls (DDT family) are all highly persistent and toxic but are also very effective and then we might discuss the implications related to their use. Given a data base and time to search it, students can uncover generalizations about DDT-like pesticides for themselves.

2) It is difficult to conduct a valuable, role-playing exercise in the Scenario BC.

Scenario AC (After Computers):

In the Scenario After Computers students still "Adopt-a-Pesticide" and gather data for a predetermined set of categories. This time, the information they collect is input (by the students) into a data base file. The students still spend two to three days "manually" finding information to complete the data fields in the pesticide record. I feel

that the practice in using the chemical reference literature is valuable and therefore do not provide a completed data base.

Each student is responsible for one pesticide record and each data field must be entered accurately, according to a set of format parameters. The format is predetermined and standardized. (see Appendix A) Careful preplanning insures that the resulting file is search and sort-able. Students enter data in pairs: one student looks on as an editor, while one student enters his/her data. By the end of the first week on the unit, the file should be complete, corrected for errors, and ready for the real fun and instructional benefit.

Week 2 of the unit is devoted to a simulation/role-playing activity. The students use the data base to make tough decisions about pesticide use to solve an insect or weed control problem. The students work in teams of three and each team is given a problem. The team must research available information, make all necessary considerations, and decide on a recommendation. An example problem for one of the teams might be:

> Problem: Gypsy Moths (Lymantria Dispar) have threatened to infest the major agricultural valley of western Oregon.
>
> Team Assignment: Your team is hired by a city pest control department to make a recommendation regarding in-city control of the Gypsy Moth. Your recommendations will be made to a city task force that is researching all options, including infertile mutant strains and natural predators. The city is interested in a careful analysis of the available pest control methods.

The students will need to complete a series of data base searches and sorts to make their selection(s) for recommendation. The searches and sorts should help them find the best insecticide, one that maximizes desirable characteristics. The desirable characteristics have been

discussed during a general introduction to chemical pest control held during the first week of the unit. The desirable characteristics are: Low persistence (length of time until it degrades into harmless substances), Non-toxicity to other living organisms, Highly effective, Low cost. An example sequence of searches and sorts might be as follows:

1) Searches
   a) Search the data base for all insecticides
   b) AND search for Pest: Moths

2) Sort and print the results of the above searches for
   a) Least to most persistent
   b) Least to most toxic to other living things
   c) Least to most effective
   d) Least to most expensive

Students can take the results of the sorts and select candidates for recommendation. This is the key information processing stage. Teams analyze and evaluate the sorts, looking for the "best" insecticide—the one that compromises the least on desirable characteristics. As students compare the results of the sorts, certain patterns or tendencies appear, e.g., Biological controls are costly but nontoxic and highly specific. Students begin to recognize these patterns with generalizations about types of pesticides. Often this will lead to additional searches and sorts. At this point, teams may need to do further research, gathering pertinent and more specific information from the chemical research literature to support their final recommendation.

Teams are required to turn in a preliminary list of considerations leading to their final candidates for recommendation. This could be the search and sort strategy they used and how they analyzed the results. This preliminary report will uncover a superficial treatment of the problem or a missing important consideration, e.g., the need for aerial

application of the insecticide makes it unacceptable to the city. The teams present their final recommendation(s) to the "public"—the class. The teams must use a visual aid during the presentation and each member of the team is required to speak.

The unit concludes with follow-up discussion of any pesticide generalizations that have surfaced as a result of searching and sorting the entire file. My hope is that the students will see certain patterns and relationships, e.g., common chemical derivatives having common methods of action or levels of persistence or toxicity common to certain compounds. By observing these relationships independently, the students are processing information and learning to make it useful.

PRESENTATION:

Using PFS File and Report, I will demonstrate the development of a data base activity for classroom use. I will discuss the major steps in the process, making suggestions and indicating problem points. The following is a general outline of the key points in the presentation.

I. Preliminaries

- Become familiar with the data base program
- Prepare students for data base use: introduce data base concepts and the data base functions; entering data, searching and sorting
- A printer is almost necessary (but don't let the lack of one stop you)
- Develop a management system for updating and storing backups of the file disks

II. Planning the Data Base and Gathering Data

- Choose the data base topic; the sky is the limit, but begin small and simple
- Preplan the format of the records in the file carefully
- As you design the file, consider the kinds of information that you want the data base to reveal
- Involve the students in the data base design
- Avoid inaccuracies as the data is "massaged" to fit the data base

III. Creating the Data Base

- Data entry is time-consuming; enlist the help of a teacher's aid, a student "clerk"--someone with keyboard skill
- Use student "editors" during data entry
- Be diligent about updating and storing a current data disk

IV. Using the Data Base to develop Info Processing Skills

- Begin questioning strategies at the recall level (develop ability to search and sort the data base)
- Progress to higher-level questioning and guide students in developing search strategies that explore relationships between data categories
- Inferring is not a trivial process--isolate and practice the steps in the process
- Encourage critical thinking--capitalize on info processing "errors" in discussion: erroneous conclusions, spurious conclusions, conclusions based on insufficient evidence

V. Follow-up

- Share and network with others creating and using data bases
- Sources for ready-made bases
- Data base topics--Your Ideas....?

APPENDIX A

PESTICIDE RECORD

TRADE NAME:
COMMON NAME:
CHEMICAL NAME:
BRAND NAME:

INSECTICIDE (Y/N): **SYNTHETIC (Y/N):**
PESTICIDE (Y/N): **NATURAL (Y/N):**
CLASS OF PESTICIDE: (Biological, Chemical, Natural)
CHEMICAL FORMULA:
PHYSICAL APPEARANCE:

TARGET SPECIES: (Names of plants, animals)
SPECIFICITY: (High, medium, low) or (Scale of 1 to 10)
PERSISTENCE: (High, medium, low) or (Scale of 1 to 10)
TOXICITY: (LD_{50} values)
EFFECTIVENESS: (High, medium, low) or (Scale of 1 to 10)

ENVIRONMENTAL AFFECTS:

METHOD OF APPLICATION: (Aerial, dust, spray)
RELATED PESTICIDES:

COMMENTS:

STATE LEVEL CURRICULUM: The Power, Influence and Opportunity

Presentation: Sandra Berry, Curriculum Resource Manager
Office of Curriculum Services
Alaska Department of Education
Pouch F
Juneau, Ak 99811

THE MAJOR FACTORS INFLUENCING OUR STATEWIDE CURRICULUM EFFORTS ARE ALIGNMENT OF CURRICULUM, INSTRUCTION AND ASSESSMENT.

- Curriculum regulations
- High school graduation requirements
- Report on the status of curriculum in Alaska districts taken from a statewide survey of curriculum directors.
- Report on curriculum models review project conducted by the Northwest Regional Education Lab in Portland.

OFFICE OF CURRICULUM SERVICES activities:

. Staffing to support local district efforts.
. Identify model programs through a process of identification called PROMISING PRACTICES and MERITS
. Support and coordinate content consortiums such as the ALASKA STATE WRITING PROJECT and the ALASKA STATE MATH PROJECT
. Provide curriculum resources
. Focus on coordination through state education organizations:
CURRICULUM CABINET
DISTRICT OFFICE CURRICULUM CABINET

DISTANCE EDUCATION

WILLIAM J. BRAMBLE, ADMINISTRATOR
OFFICE OF INSTRUCTIONAL SERVICES
ALASKA DEPARTMENT OF EDUCATION

WHAT IS DISTANCE EDUCATION?

The distance education proposal consists of a new support system for the delivery of courses for high school students in Alaska. The concept builds capacity for local schools through the use of new and existing educational technologies including television, computers, audioconferencing, and teletext in conjunction with the classroom instructor.

WHAT IS THE RATIONALE FOR DISTANCE EDUCATION?

I. Distance education would help assure equality of access to educational programs for all students in Alaska.

II. The model of education in high schools was developed decades ago to meet the needs of that time period. Experience shows that standard models of delivery do not necessarily meet the needs of today's students, particularly those in rural settings.

III. Distance education helps provide the small school the opportunities to build on its inherent strengtns

Some of the inherent strengths of the small school are:

1. Greater opportunities for the small school becoming learner-centered.

2. Increased potential for effective individualization of instruction and for broad independent study programs.

3. A greater sense of community and participation felt by everyone in the school; smallness by its nature creates involvement.

4. One-to-one relationships between teachers and students botn in the classroom and in the community.

5. Cross-grading of students in classes resulting in a more stimulating age mix--one that approximates more normally the age mix of people in society.

6. A schedule that can be altered more readily to take advantage of special events, field trips, workstudy programs, etc.

7. Closer coordination among teachers, resulting in a more unified approach to goals and problems.

8. Close working relationships between the small school and the community, which can cultivate mutual respect and understanding.

III. Teachers and administrators often find it difficult to meet the needs of students in rural Alaska high schools where special conditions exacerbate the inherent flaws in the traditional model and create difficulties in assuring equality of educational opportunities.

Some of these conditions are:

1. The small school's enrollment and staff size make it difficult to offer broad and varied courses.

2. The small school and rural communities often have limited support services--e.g., available resources in the community such as laboratories, museums, tutors, libraries, etc.

3. Small class size in many courses allows for little student interaction or peer tutoring.

4. Difficulties in staff recruitment and retention are often associated with isolated settings. Multiple preparations, housing situations and similar factors contribute to this situation.

5. Limited alternatives are open to a student and teacher in cases of incompatability of personalities or learning styles. A student generally cannot be moved to another section of the same course in a small school if only one section is offered.

6. The necessity for teachers in small schools to teach multiple subjects and grade levels sometimes results in teaching assignments with teachers working outside their areas of specialization.

7. Difficulties sometimes exist with access to courseware and teacher training designed to meet specific and unique needs of rural Alaskan students.

There is a need to establish school patterns and programs appropriate to small schools rather than replicating the processes and procedures of larger schools. Small schools are attempting to do this throughout Alaska. Although there are many aspects to this issue, the proposed distance education program addresses this need in some significant ways.

HOW DOES DISTANCE EDUCATION HELP FILL THIS NEED?

Distance education provides the small school the opportunities to build on its inherent strengths while alleviating some of the pressures of the conditions previously noted.

These strengths are an excellent base upon which to build. The increased resources to be offered to small schools through distance education can assist in emphasizing these strengths and minimizing weaknesses.

WHAT CAN BE OFFERED THROUGH DISTANCE EDUCATION?

1. High-quality courseware could be developed for grades 10-12 and be made available to students in public schools statewide Thirty course offerings are proposed that would be available at local option.

2. Access for students and on-site teachers to instructors who are specialists in the particular content areas.

3. Tutorial assistance both from the distance instructor and in the form of peer tutoring from students in the distance classroom.

4. Interaction with other students enrolled in the same course. Discussion and problem solving sessions with students comprising a distance classroom.

5. Access to high quality print and media materials. Each course would provide excellent materials designed to optimize learning opportunities.

6. Appropriate computer assisted learning. The power of modern computer technology would be used in forms most appropriate to each course of study.

7. High level academic courses for students that build entry skills to post secondary education and/or employment.

8 Access to broad information resources and data bases would be provided through a variety of means to reduce isolation and overcome the problems of limited resources in small communities.

HOW WILL IT WORK?

At the option of local schools, courses offered through the Distance Education Program could be selected for instruction. The courses would include core academic courses in mathematics, science, language arts, social studies, foreign languages and other areas. For any course, the school could choose to use the course materials and learning activities within the class framework or could choose to have students participate in a distance classroom. The distance classroom would consist of students at a number of schools receiving daily interactive instruction from a distance instructor with special training and expertise in the subject area.

Students would take courses in the local school setting under the direct supervision of on-site teachers who would play a critical role in the instructional process. Distance education offerings would both assist the teacher and enhance learning opportunities for students.

Significant features in the student environment would consist of the following:

*Classroom Computers for drills and practice, simulations, problem solving, tutorials and word processing.

*Data Terminal for access to the distance teacher, access to other students, access to information bases, testing, homework assignments.

*Television Monitor for direct instruction, interactive programs

*ITV Programs for current events, access to specialists.

*Audio Conferencing Equipment for tutorials, group discussions, peer interactions and tutoring, direct instruction.

*Teletext for high resolution graphics used with live broadcasts, delivery of homework and tests, instructional activities, and access to specially constructed instructional data bases.

*On-Site Materials including textbooks, manuals, lab equipment, and all other instructional materials required for the instructional process.

HOW WILL THE CLASSROOM TEACHER BENEFIT FROM DISTANCE EDUCATION?

The classroom teacher will be able to better serve each of his/her students. With distance education among the options available, the teacher becomes a facilitator for the learning process of each student. It will be much easier to individualize instruction and tailor each students progress to his/her needs. Distance education will reduce the pressure of the classroom teacher to be a specialist in several curriculum areas. The teacher can concentrate on his/her strengths in facilitating and supporting learning, rather than trying to be knowledgeable in all areas. The benefits of a small school can be more fully realized by assisting classroom teachers in these ways.

HOW WILL STUDENTS BENEFIT FROM DISTANCE EDUCATION?

Rural teachers will have powerful instructional tools available to assist them in instructing students. Students will be able to take courses not readily accessible to them due to small staff size or limited facilities. Students in small schools will have a wider range of academic subjects from which to prepare themselves for college or the world of work. Students will be able to interact with others enrolled in the same distance course, thus expanding their opportunities for peer interactions and often expanding their cross-cultural experiences. Rural students could be given opportunities comparable to urban students for quality, varied course offerings. Course materials and services will also be available for use in urban schools upon request. It is anticipated that in specialty course areas, urban school would find the materials helpful for providing quality instructional programs.

WHEN COULD DISTANCE EDUCATION BE AVAILABLE FOR STUDENTS?

The proposal is in the first stages of development. If approved by the legislature, it would be pilot tested prior to full implementation. Local district participation will be sought in designing the services as soon as funding is obtained. The curriculum and the specifics of instruction would be defined by a group of local educators. If implemlented fully, within five years, teachers and students would be able to choose from among 30 course offerings. Tne timeline for full implelmentation is:

Fiscal year 1986--Completion of the delivery design, and initiation of development of six courses.

Fiscal year 1987--Pilot testing of model, six courses completed, six additional courses under development, support capacity developed, and local site development and training.

Fiscal year 1988--System initiated with 12 courses on-line, an additional 12 courses under development.

Fiscal year 1989--24 courses on-line, six courses under development.

Fiscal year 1990--System fully operational.

SUMMARY

Teachers and administrators in all parts of Alaska are working hard to create a quality education for their students. Unfortunately, their hard work and dedication cannot overcome all of the challenges presented by rural education without special assistance. The unique combination of factors confronting educators in this northern-most state demand a considerable degree of expertise and innovation in tne development of secondary programs.

One aspect of the solution of problems confronting secondary education in this culturally diverse and physically vast environment is the availability of several educational options. Those options can tnen be selected and used together or separately to individualize a student's education. Distance education can be an important addition to the educational delivery system in the state. It can offer students the opportunity to more fully benefit from a nigh school education regardless of where they live.

KEYBOARDING: HELPING STUDENTS LEARN ABOUT LEARNING

C.M. Chmielowski
Fairbanks North Star Borough School District

Our middle school decided to teach our 350 seventh graders touch-typing, using a CAI program on Apple IIe microcomputers. This paper describes two aspects of our experience. First, it discusses the mechanics of setting up an alphabetic keyboarding program in a middle school/junior high. Second, it looks at how this experience could be a model for the use of other kinds of CAI.

The 750 seventh/eighth graders in our school are grouped in teams of 125 students, with 5 teachers assigned to each team. The school has a computer lab with 24 Apple IIe's, staffed by a full-time computer co-ordinator. The school's computer applications committee, made up of members from each department and team, makes recommendations to the co-ordinator on scheduling and software used in the lab.

Last year, many of the teams brought their students into the lab for a week or two to use the computers as word processors. They were delighted with the response of the students to being able to write, revise, and print with the computers. But they were disappointed with how slow the students typed.

To see how the computers might be effective tools for teaching typing, the typing teacher and the computer lab co-ordinator spent several weeks looking at the research already done, and experimenting with various typing programs

and different groups of students. After we told the committee our results, they decided to purchase a lab set of Alphabetic Keyboarding, which is a two disc program distributed by South-Western Publishing Company (855 California Avenue, Palo Alto, CA 94304)

We started using these discs at the beginning of this year. Every seventh grade team came into the lab for three weeks, using this program daily in 40-minute periods to learn typing. Each worked at their own computer, and moved through the material at their own pace. The time for this work was taken out of the curriculum in one of two ways. In one team, the students came in place of their reading class for three weeks. In the other two teams, the students rotated through their five team classes, coming out of a different class each day. Thus they only missed one day a week in each of their classes. At the conclusion of these three weeeks, they spent several additional days in the lab using a word processor and other applications software.

We came to the following conclusions on the mechanics of teaching keyboarding using micro-computers.

1. Students need at least five - not three - weeks to gain sufficient skill so they will use touch-typing instead of the "hunt-and-peck" method. Three weeks was enough time for only the very quickest students to completely learn the alphabet.

2. South-Western Publishing produces an excellent program. Each disc comes with a back-up, and the publishers

were prompt to replace at no charge any discs with problems. After the large group discount was applied, the price was very reasonable.

3. The support of the typing teacher and all the classroom teachers was essential. They offered valuable suggestions in the initial planning and throughout the implementation. Furthermore, because they were involved in the decision-making process, they suppported the project, even though - as with any new project - there were many bugs to be worked out.

4. Using typing games was only helpful for students who had already mastered the keyboard and wanted to build additional speed. Other students were either frustrated, or reverted to their "hunt-and-peck" technique to prevent avoid death by alien invasion.

5. The maximum period of intense concentration using this CAI was 30 minutes. Fifteen to twenty minutes would probably be better for most students.

6. An effective way to use the lab time after the students had completed their day's keyboarding work was to expose them each day for 10 or 15 minutes to a different program or peripheral that could be used on the computer. Once they were aware of these possibilities, they would often come in during lunch and before or after school to develop skill in using an application that appealed to them.

In general, we were satisfied with the results of using this program, and intend to use it again next year

with all the 7th graders. But the most interesting result came from an area not related directly to keyboarding. Those observations will be the focus of the rest of this paper.

After the first group of seventh graders had completed their time in the lab, we were only moderately satisfied with the results. So, before the second group of seventh graders came, we invited the district's study skills co-ordinator, Bev Byington, to offer suggestions for improving the class. She had spent a year's sabbatical leave studying the techniques used by successful businesspeople to set and attain their goals. She helped us devise a plan for the second group of seventh graders which incorporated these techniques.

The results were deeply satisfying. The students indeed learned more keyboarding skills in a shorter period of time (Fig. 1). But they also began to learn such things as:

techniques of analyzing and evaluating their own learning;

techniques for motivating themselves to achieve a desired goal;

the value of working with peers as a way of increasing their own rate of learning;

how goal-setting relates to success.

How did this happen?

First, before they ever came to the computer lab we

spent a period drawing out of them reasons why they would want to learn to touch-type (goal-setting). As the students gave their reasons, we wrote them, exactly as stated, on butcher paper. The butcher paper was then posted in the lab for as long as their team was in the lab. This time spent encouraging them to "buy into" the goal of learning keyboarding resulted in significantly more commitment to the task.

Another technique that proved very useful was to give all the students a list of "steps to success" (Fig. 2). We wanted them to have a set of steps which, if followed, would guarantee they would meet their goal. From the experience of the first group, we culled comments about what helped them to learn keyboarding efficiently. We gave this list to the second group before they came to the lab. Then the students continually reinforced and added to this list. Each day we asked them to share which of the "steps to success" had been most helpful to them that day. We also asked them to share with the class any additional steps to success they had discovered for themselves. This encouraged them to reflect daily on how their own learning was succeeding - to evaluate the steps they were taking to be successful learners.

Sharing their evaluations of their learning success was one way they contributed to each other. Another way was to test each other following each lesson. A student would hold a sheet of paper over the keyboard of the person next to

them, and watch the screen to see if the passage were typed correctly. S/he would then sign their name on their partner's personal success chart, certifying that the partner had passed the test. (We were afraid that this would cause considerable disturbance; in fact, they became very adept at testing one another without distracting those around them.) This helped create a co-operative attitude, and encouraged communication between students on how to improve keyboarding skills.

The use of the "personal success chart" mentioned above (Fig. 3) was another way to help students take charge of their own learning. Each student was able to monitor daily their own success by keeping a chart on which they signed off each lesson as they successfully completed it, and had a friend sign it off . Seeing the sequence of objectives always in front of them helped them see how their goal could be broken up into a number of objectives, each of which would be possible to achieve. This encouraged them to be responsible for deciding what to learn each day, and work to move towards that goal.

Motivating them to work silently, consistently, at developing typing skills was not nearly the problem with the second group as it had been with the first. Once they were convinced they could indeed be successful by followin the "steps to success," they did not want to STOP working. The motivation of simply wanting to be successful at a task proved extremely powerful.

While the students were improving their ability to understand, analyze, evaluate, and share their learning, their teachers were discovering ways of working with students that encouraged this kind of student work. Since the computer was providing instruction and drill and practice on an individualized basis, the teacher had time to do other things.

One of the most enjoyable things we did was to devise and present certificates at the end of each class to those who had achieved a certain level of competency. We also had the chance to walk around the room, speaking with each student once or twice quietly during the period, commenting on something they were doing well. We could spend time in individual conference with students having a particular problem, and direct them to the resources (usually one of the "steps to success") which might help them solve their problem. We could work with students to help them analyze and solve their problems, becoming co-learners with them. We could spend time devising with them with ways of organizing their time and efforts to increase the effectiveness of their learning. Because students were feeling successful about what they were doing, they remained on task, and discipline problems were almost non-existent. From a teacher's standpoint, it was a most desirable environment.

This experience using a highly structured CAI program, alphabetic keyboarding, can serve as a model of what will be increasingly possible in schools, as good CAI for a

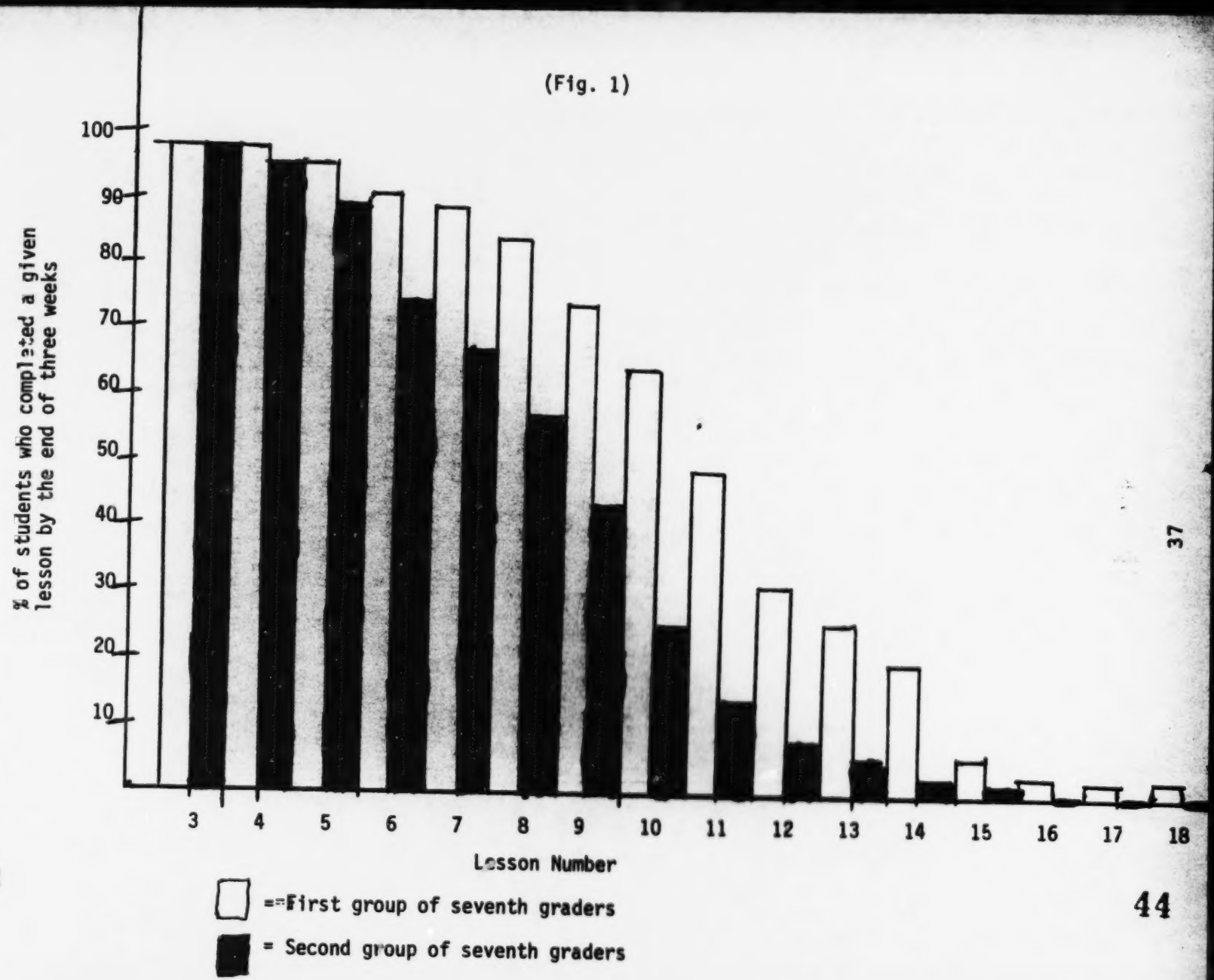
variety of subjects, and inexpensive computer hardware for every student become a reality. With CAI in other subjects, the "steps to success" would of course be different. The ways that peers could help each other would change. But the use of CAI as a way of encouraging students to develop higher-level thinking skills, to take responsibility for their own learning, to work co-operatively with their peers, could be the same.

Sometimes a distinction is made between two uses of the computer. It is a tool for tapping and developing the higher human abilities of creative problem-solving, goal-setting, analyzing, evaluating, and communication through such such uses as word-processing, graphics programs, and Logo. But CAI, using tutuorials and drill-and-practice, is seen to address relatively lower cognitive skills, such as memory and understanding. Our experience suggests that this is not a necessary distinction. The teacher makes the critical difference.

The teacher can structure the learning environment so that good CAI can be the vehicle for students to develop goal-setting, creative problem-solving, analysis, evaluation, and peer-interaction skills. CAI is not incompatible with developing these abilities. Furthermore, if the teacher does this, it will turn out to be a much more pleasant environment for the teacher. It can be a "No-Lose" situation, where the students master the desired skills, develop higher-level cognitive skills in the process, develop the tools needed for

life-long learning by learning about how to learn, and students and teacher work together to create a pleasant, eu-stressful environment for all.

(Fig. 1)


% of students who completed a given lesson by the end of three weeks
100
90
80
70
60
50
40
30
20
10
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Lesson Number
=First group of seventh graders
= Second group of seventh graders

43

44

(FIG. 2)
STEPS TO SUCCESS

1. Work in SILENCE.

2. When you are frustrated, stop. Admit you are frustrated. Breathe deeply a couple of times. Look at the charts on the wall on which you have written why you want to learn keyboarding. Read through these "Steps to Success." Know that you will continue being successful on the computer after you take a break.

3. Read each screen on the computer carefully before presssing return to go on to the next screen.

4. Work through each section of the lesson in order, not skipping any, until you reach the point where the message appears, "Congratulations! You have finished this lesson."

5. After you have finished a lesson, and feel successful with it, go back to the section of that lesson which your success chart tells you to be tested on. Type two lines that contain the new material, with no more than one mistake. Then sign your name by that lesson on your success chart.

6. Have a friend or a teacher hold a paper over your fingers and watch you type two lines that contain the new material, with no more than one mistake. Then have that person sign their name by the lesson on your personal success chart.

7. When one particular letter frustrates you, return to the section that teaches that letter, and practice that section until you feel successful.

8. Work slowly, to be able to eventually work fast.

9. Anything worth doing is worth doing poorly. Making mistakes means you are trying to do something new and worthwhile.

10. Keep your eyes on the screen at all times.

11. Keep your fingers always on the correct home-row keys.

12. Feel the raised dots on the "D" and "K" keys when you need to find your place.

13. Know that you CAN do it!

(Fig 3)

PERSONAL SUCCESS CHART FOR

LESSON NUMBER	SECTION TO BE TESTED ON	LETTERS LEARNED IN THIS LESSON	GWAM	SIGN-OFF BY YOU	SIGN-OFF BY FRIEND	SIGN-OFF BY TEACHER
1	6	A S D F J K L ;				
2	5	Review				
3	8	E H O				
4	8	T I Left Shift				
5	6	Review				
6	6	Review				
7	8	N G Right Shift				
8	4	Review				
9	6	U C . P				
10	7	Y , X				
11	3	Review				
12	7	M Z				
13	7	B ? Q				
14	4	Review				
15	5	V				
16	3	Speed				
17	3	Speed				
18	4	Speed				

SPEECH SYNTHESIS FOR EARLY ELEMENTARY

Scott Davies, Computer Coordinator, Klawock City Schools

In many educational content areas, microcomputers have served as relatively low-cost tools for interactive drill-and-practice activity.

Unfortunately, in the area of early reading skills, this tool falls significantly short of what it is capable of doing in other content areas, due to the fact that text is ordinarily the only in-depth means by which the computer and student may interact.

To get around this seeming obstacle, speech capability may be added to a computer by means of a speech synthesizer unit (which may be either a plug-in board or "stand-alone" unit).

Computer-synthesized speech may be accomplished by use of one or more of the following four techniques:

1) English-text-to-speech conversion -- In this, the speech unit (via either software or firmware) examines the text to be spoken and attempts to pronounce it using English pronunciation rules.

   While this is the most flexible of the strategies and permits virtually any word to be spoken, it tends to be less "life-like" and will occasionally mispronounce words.

2) Phoneme stringing -- This approach is similar to the "text-to-speech" approach in that text contained within the program is examined by the computer and then pronounced according to pre-established rules. This time, however, the pronunciation rules are not those of the English language, but rather, an artificial code established by the manufacturer.

This ends up sounding similar to text-to-speech, but greater control over sounds is possible. It is, however, more difficult to program, and re-writing is required for software conversion.

3) Word stringing -- In this mode, words to be pronounced must be selected from a collection of words provided on disk.

The quality of individual words is improved over the previous two techniques, but intonation may be lost (i.e. -- the word may always pronounced exactly the same, regardless of its placement), and is limited to words provided by manufacturer.

4) Custom speech mode -- This has the capability of delivering the best-sounding speech (depending largely on the talents and degree of effort put into it by the programmer), and may be used to speak virtually any pronounceable word.

Unfortunately, it is also the most difficult to program. If speech software of this nature is available for the speech unit being utilized, great; if it isn't, it can probably be ruled out as being beyond the capabilities of the "do-it-yourself-er".

CURRENTLY-AVAILABLE SPEECH SYNTHESIS UNITS:

Following are SOME of the commercial speech synthesis units currently available for the Apple][series, listed by manufacturer.

NOTE: While some units may be available for the //c, the expansion slots of the][+ and //e make them more likely candidates).

1) Street Electronics Corporation
1140 Mark Ave.
Carpinteria, CA 93013

(805) 684-4593

-- Echo][. Price: Latest list is about $150 (ours were $99 each).

This unit is one of the lowest-priced available, and can utilize any of the synthesis techniques described above. In the "text-to-speech" mode, quality is fair, sounding much like the ""Cylon"" voices in "Battlestar Galactica", or the TI "Speak-and-Spell" toys (which also uses TI's TMS 5200 speech chip).

Some educational software is available from third-party vendors.

-- Cricket. Price: Estimated $150-200.

To my knowledge, the only unit designed specifically for the //c. It seems to share many features with the Echo][, and adds music capability and a clock.

2) The Micromint, Inc.
561 Willow Ave.
Cedarhurst, NY 11516

(516) 374-6793

-- Lis'ner 1000 (base unit). Price: $189 (assembled), $149 (kit)

-- Lis'ner 1000 (w/text-to-speech). Price: $259 (assm), $219 (kit)

These units differ from the others in that they also offer speech RECOGNITION (the ability to "hear", via a microphone, what the user is saying). While this is a promising feature for handicapped students who cannot utilize a keyboard, the field remains somewhat crude at this point.

-- Sweet Talker I. Price: $89.

Uses text-to-speech or phonetic code. Built around Votrax SC-01A chip.

-- Sweet Talker II. Price: $104.

Improved version with added music capabilities. Uses SSI 263 chip.

-- Microvox. Price: $349 (assembled), $269 (kit)

Special emphasis added to text-to-speech capabilities. Is essentially a "stand-alone" peripheral, having its own power supply and containing all code in firmware.

3) Jostens Learning Systems, Inc. (formerly Borg-Warner)
600 W. University Drive
Arlington Heights, IL 60004-1889

1-800-323-7577

-- Ufonic UVS. Price: $495

-- Ufonic UVS-II (headphone output only). Price: $350

This unit has the best-sounding speech quality of units that I've heard; however, it appears to lack a true text-to-speech conversion mode.

Unlike competing units, the software utilities required to "home-brew" software are not included in the base price, and must be purchased separately at a cost of $225. Fortunately, Jostens offers a good variety of educational software.

4) Votrax, Inc.
500 Stephenson Highway
Troy, MI 48084

(313) 588-0341

-- Votrax PPS. Price: $189

-- Votrax TNT. Price: $395

Votrax was one of the earliest speech synthesizer manufacturers. Their literature on the above units states that they utilize a phonetic approach (which seems to imply either a text-to-speech or phoneme-based approach, or both).

They are currently conducting a contest for "best application ideas", which suggests that they intend to be producing their own educational software.

COMPUTER SCHEDULING

APPLE 11+,11e vs IBM SYSTEM 38

Mike Dishnow & Bonnie Murrill
Matanuska-Susitna Borough Schools

Three Matanuska-Susitna Schools have been using computerized scheduling systems over the past several years. Wasilla High School initiated the process with the CHARLES MANN PACKAGE, THE ASSISTANT PRINCIPAL, during the 81-82 school year -- the initial run was to schedule a career fair that Spring. The fall of 82 was the initial computer scheduling of the WHS student body classes. WHS had a student body of 700+ in 82-83 and grew to 1,200 by the fall of 85; Charles Mann was used throughout this period. Wasilla Junior High School and Palmer Junior High School rapidly followed suit. Wasilla Junior High School, the larger of the junior highs, grew from 450+ to 800+ under Charles Mann. Charles Mann is a micro-based system -- it will function on the 48K Apple 11+ or the 11e.

The IBM System 38 was installed by the school district in the Spring of 84. The STUDENT ACCOUNTING PACKAGE was purchased from J & K COMPANY, Salt Lake City, Utah. This package includes attendance and grading modules as well as scheduling -- initial indications point toward the latter two, attendance & grading, as being the most powerful and useful. Palmer Junior High School piloted the STUDENT ACCOUNTING software during the fall of 84. Wasilla High School used the scheduler for the Spring 85 semester and brought attendance on line concurrently.

THE PURPOSE OF THIS PRESENTATION WILL BE TO SHARE THE MAT-SU EXPERIENCE AND TO DEVELOP A COMPARISON/CONTRAST BETWEEN THE SMALL VS LARGE SYSTEM SCHEDULERS. An Apple 11e will be set up to allow participants to look over the micro-based Charles Mann software following the presentation.

Pages two and three of this abstract will provide an outline and allow space for participants to take brief notes.

APPLE 11+,11e SYSTEM 38

FACTORS

Decision to buy:
(level of)

Cost:

Documentation:

In-service/Training:

Implementation:

1. teachers -
2. students -
3. counselors -

Competent Support (THE KEY!!):
Secretaries

Data Gathering & Input:

Run (Rerun) Time:

Print Time:

Error Rates:

Versatility/Products:

Updating:

Interface:

Program Bugs/HELP!!:

Custom Programming:

THE MOST IMPORTANT CLOSING COMMENTS ARE THESE:

1. **INSIST ON ADEQUATE DOCUMENTATION.**
2. **TALK WITH SCHOOLS/DISTRICTS ALREADY USING THE SYSTEM(S) YOU'RE INTERESTED IN (BEFORE PURCHASING).**

First Level Mathematics (KINDERMATH II)

Mary A. Felleisen, Project Director
First Level Mathematics (Kindermath II)
PRIMAK Educational Foundation
Devon, PA 19333

This presentation includes the demonstration of a 13-disk software system that is appropriate for children in their first year of mathematics instruction. The software is based on nationally validated printed materials for kindergarten and first grade that provide a sequential curriculum and management system for teaching basic mathematics skills.

The software is diagnostic/prescriptive in nature, with the child s entry level skills being determined by an assessment disk. From there the learner begins to proceed through as many as 90 step-by-step closely sequenced objectives that teach the broader areas of same/different, patterns, sets 0-10, shapes, matching numerals to sets, math signs, addition and subtraction. Each of the objectives is presented in the form of a lesson, with an introduction, teaching activities and an assessment before the student may automatically move to the next step. The lessons branch, re-teach or review the previous activity based on the child's progress within each succeeding lesson. A built-in management system provides for complete record keeping whether in a classroom or lab setting. Childrens' dates of participation, lessons learned, specific problem areas are all recorded and stored on the teaching disks. They may also be printed out or stored on a data disk for future reporting to parents.

Developed by a team of educators and programmers, the software system uses a joystick instead of a keyboard. Children are given instructions by means of a voice-synthesizer. The colorful graphics introduce mathematics concepts without requiring any reading skills, thus making it especially appropriate for not only kindergarten and first grade children but in programs of remedial and compensatory education. The interactive, multi-sensory approach of voice, high-resolution graphics and joystick response is particularly useful for students with special needs in such areas as fine motor-coordination, visual or auditory acuity. The material is child-centered with built-in praise and success.

The printed version of the materials and the accompanying manipulatives exactly parallel the lessons on the disks, and have been shown to provide educationally significant growth in young children, relative to national norms on a standardized mathematics instrument. The presentation will not only provide participants with an opportunity to review the program in both its software and print format, but also be apprised of the training and technical assistance services that are available to educational agencies.

THE USE OF PROGRAM EVALUATION AND REVIEW TECHNIQUES (PERT) IN EDUCATIONAL ADMINISTRATION

Dr. Thomas A. Gaylord
Assistant to the Executive Vice President
University of Alaska
Statewide System of Higher Education

INTRODUCTION

With the evolution of a technologically complex society, traditional methods of planning, decision-making, and problem solving have become inadequate tools for the educational administrator. It has become evident that exclusively verbal descriptions of complex systems/activities and their interactions have resulted in generalizations which have presented problems for comparison, analysis, and evaluation. Numerous studies have documented the increasing importance of quantitative methods in several areas of administrative endeavor (Banghart, 1969; Bertalanffy, 1956; Churchman, Ackoff, & Arnoff, 1957; Goodlad, O'Toole, & Tyler, 1966; Hall, 1962; Optner, 1965; Van Dusseldorp, Richardson, & Foley, 1971). Three principle areas in which quantitative techniques have been applied include: (1) planning, (2) organizational control and the decision-making process, and (3) the analysis of how organizational systems work and how their different parts or subsystems are interrelated (Bell, 1977; Martino, 1972; Optner, 1975).

A number of quantitative approaches have evolved that apply the concepts of general systems theory. The major approaches stated by Immegart and Pilecki (1973) have included: (1) cybernetics, (2) holism, (3) output analysis, (4) systems design, (5) information theory, (6) systems analysis, (7) systems engineering, (8) mathematical programming, (9) computer science, and (10) operations research. Utilization of an operations approach was made in this study to facilitate the planning and budget process at the higher education system level.

PROBLEM/PURPOSE

The need for improving the coordination and completeness of system level research project/data requests to institutional units to satisfy increasingly complex and detailed internal and external information demands can best be illustrated by the annual planning and budget schedules used by higher education institutions. In many cases, the schedules are nothing more than calendars of due dates that do not allow measuring the impact of adding a new data requirement or adapting to a change in budget reporting requirements from the state. By treating the planning and budget cycle as a single large project, and then applying networking project management techniques, the intended outcomes would include:

1. Forcing planners to specify in detail the set of activities which constitute the project and to establish clear project responsibilities.
2. Clearer delimitation of major project problems.
3. Easier monitoring of project status.
4. More effectively communicating the project tasks and resource interdependencies by using the networking model graph.
5. Allowing easy testing of alternative scenarios.

This paper presents the results of an initial research effort in the development of an operations research networking model intended to improve the planning, organizational control, and decision-making processes at the higher education system level. The PERT-type model provides a quantitative mechanism allowing description of resource interdependencies that exist among statewide and unit offices and the generation of multiple scenarios for the analysis of "what if" questions by interchanging available resources, relaxing the technical specifications of the model, and changing the arrangement of required tasks or activities.

METHODOLOGY

A statewide university planning-working team composed of key administrators from the academic, fiscal and facilities planning areas was assembled to outline project strategy and begin collecting system level data to answer the following questions:

1. What project planning activities are required?
2. What are the sequencing requirements or constraints for the activities?
3. Which activities can be conducted simultaneously?
4. What are the estimated time requirements for each activity?

A PERT-type diagram was constructed using Lisa Project, a project management software tool that runs on an Apple Lisa microcomputer. The program generated six charts (a PERT chart, a resource Gantt chart, a task Gantt chart, a resource costs chart, a cash flow chart, and a project table) which were presented and distributed to academic and administrative unit heads for review before completion.

RESULTS/IMPLICATIONS

It was possible to include all major academic, fiscal, and facilities planning activities required at the system level in the PERT model. The final model was comprised of over 110 separate tasks and included over 70 milestones. It included legislative, gubernatorial, governing board, and university system level milestones and displayed what specific system level and campus level offices were required for each task.

The major strength of the model has been simply in organizing the overall planning and budget effort. It also allowed for testing alternative plans, revealing the real dimensions and complexity of the planning and budget project, establishing well-understood management responsibilities, and identifying realistic expectations for the project and the task constraints used. The questions the model helped answer included:

1. When will the planning project be completed?
2. When is each individual part of the project scheduled to start and finish?
3. Is it possible to shift resources to critical parts of the project without affecting the overall completion time of the project or critical meeting dates?
4. Which tasks are time critical to the rest of the project?
5. Where should management concentrate its efforts at any one time?

Model weaknesses include the fact that stochastic processes were not considered and a manual review of the model is necessary if two tasks overlap and only one at a time can be performed by an office.

Turbo Pascal in the High School Computer Lab

Gary N. Gustafson
Computer Science - Wasilla High School
Mat-Su Borough School District
Box AB, Palmer, Alaska 99645

Hey - who needs it anyway? - We all have LOGO and BASIC!

A review of a specific implementation of any computer language assumes that there is good reason to use that language in the first place. The extended BASIC language implementations provided with microcomputers today are powerful and fast, and LOGO seems to be easily integrated into curriculums at any grade level, so why would anyone choose to use or teach Pascal?

Reason 1 - to teach programming:

Pascal is a language designed by Niklaus Wirth in the late 60's specifically to teach computer science and programming. It is designed to force the student to write programs that are well documented with a logical structure that is clear and readily identifiable. Wirth patterned the language after the structure of ALGOL 60, a computer language used widely in Europe; the result is a language that is not only well suited for education, but is also powerful and elegant enough for commercial applications. It is of the procedural family of languages which includes LOGO, Lisp and the popular C language. Procedural languages do not depend on line numbers, as BASIC does, but group programming code into related blocks called procedures and functions. Each procedure or function is identified by a name. The programmer merely needs to include the procedure name in the program wherever that particular block of code is to be executed. Suppose, for instance, that we want to be able to change a sentence from lowercase letters to all capitals (maybe we're getting ready to print mailing labels - the Post Office likes capital letters). We could write a procedure in Pascal to perform that chore; we'll label that procedure UPPERCASE. We then will merely refer to UPPERCASE, followed by a

reference to the sentence, whenever we need to change a sentence to upper case letters. Simple - and clear. No obscure GOSUB or GOTO. No wondering what happens at line 5000. But, LOGO is like that too; so why not use LOGO instead?

Another strength of Pascal as a language to teach programming skills is the clearly defined way that data structures are handled. All computers store information in ways that are very specific on the machine language level. Information must be stored in such a way that different types of information can be handled efficiently and without ambiguity. Numbers used in computations are different from numerals used in street addresses or telephone numbers, and from groups of characters that form words or sentences. Integers used for counting are usually stored differently from numbers that represent decimals (Real numbers). An understanding of the differences in the way these varieties of information types are stored and handled is vital to the efficient programming of any computer. Many programming problems cannot be solved without such an understanding. LOGO, and some other higher level languages, are somewhat ambiguous in the way various data types are handled. Pascal is designed to clearly delineate the variety of data types. All variables used within a Pascal program must first be identified, or "declared", as belonging to a specific data type. Pascal has five standard data types: Integer, byte, Real numbers, Boolean value (logical true/false), and characters. It also has specific rules for combining these data types into more complex types for any application. Although Pascal is truly a higher level language, it is designed to allow exploration and use of structures at the machine language level. Students learning Pascal gain a clear understanding of how information is being stored and manipulated within the computer.

Pascal's similarity to other programming languages currently being used to develop commercial programs is another strength. The C programming language is the best example. C is used to develop programs that must be fast and efficient:

such as operating systems, word processing programs, and integrated software. Pascal and C are very much alike in the way they use procedures and functions, and in the way they treat data types. The words used in C tend to be a bit obscure, and the language is a bit harder to learn than Pascal. Pascal uses simple, clear labels for its standard functions and procedures, and is relatively easy to learn. It is a good first language that can easily lead to C or any other procedural language.

Code written in Pascal is much easier to understand than code written in BASIC. The Pascal programming style and nature of its data structures encourages writing code that looks like english phrases. As a result, Pascal code is self documenting to a large extent. Students find it much easier to go back to an old program and pick up where they left off. Example programs are also much easier to follow and understand.

Students tend to write programs that are more clearly structured and better defined logically when they program in Pascal. They gain a better grasp of how to structure a program to solve a particular programming problem; and they are less likely to get lost in the minor details of the program. Complex programs, which tend to become a tangled mess in BASIC, or cumbersome (or too large for memory) in LOGO, can often be written easily in Pascal.

Reason #2 - to develop programs for classroom use:

Pascal programs are usually much easier to maintain and change than programs written in BASIC or LOGO; especially as the programs increase in complexity. The programs are written using a text editor or a word processor and later compiled into native machine code or UCSD P-code for use on the computer. Creating and editing a program has all the advantages of using a word processor for any document. The entire program can be examined continuously on the screen.

Pascal programs usually are faster than programs in BASIC or LOGO. That,of course, is a generalized statement. BASIC and LOGO both have specific features that tend to make them better suited for some applications - especially when they are designed to make direct use of a specific computer's unique features. However, Pascal is a compiled language that results in a machine language or P-code file, and BASIC and LOGO are interpreted languages which result in code that must be translated into machine language every time it is used. Compiled languages are generally much faster than interpreted languages. Although the BASIC provided with the computers in the lab at Wasilla High is extremely powerful, and designed to make full use of the computer, I generally prefer to develop educational programs in Pascal due to the efficiency of the resultant program and the ease with which programs can be altered or adapted for other uses.

Reason #3 - it's available for most microcomputers!

The lab at Wasilla High is made up of 20 Corona PC computers (IBM PC compatible). There are several implementations of Pascal available for this type of computer, ranging from a cost of around $50 up to over $500. Several good implementations of Pascal are available for virtually all microcomputers

OKAY - so what about TURBO PASCAL?

TURBO PASCAL is an implementation of Pascal by Borland International of California that retails for $49.95. Almost anything would be worth purchasing at that price, but this Pascal is capable of developing highly sophisticated programs at a commercial level. Although inexpensive, it is not a toy. It is an almost complete implementation of standard Pascal (the omissions are very minor). It provides several extensions to Pascal, such as enhanced string handling procedures and functions, which are also found in other Pascal implementations. Documentation

is in the form of a well arranged and indexed soft-bound book of over 300 pages. Although it is not a tutorial, it does a very good job of explaining how the various commands, gives detailed descriptions of the data structures, and includes helpful sample programs and routines. It's well above average for most software documentation and is a good reference for someone who already knows the language..

TURBO will run on computers with the following operating systems:

Operating system	Microprocessor	Examples of some computers
CP/M	Z-80 only	Osborne, Kaypro, Apple (with CP/M card)
PC-DOS	8088/86 family	IBM PC, IBM PC jr & compatibles
MS-DOS	8088/86 family	Corona PC, Compaq, Tandy 2000
CPM-86	8088/86 family	

Further extensions are included in the language to allow direct manipulation of the operating system calls and features, much as an assembly language programmer would do, but with greater ease.

Perhaps the best feature of TURBO PASCAL is that it combines the functions of the program editor and the compiler. This eliminates the common frustration of having to switch from a text editor to a compiler, and sometimes to an assembler, when working with compiled languages. In Turbo, code can be written, compiled, executed, and tested for bugs, all from a single unified program. If that weren't enough, TURBO also returns to the written source code to indicate errors detected during compilation or execution. In many ways, TURBO is almost as interactive as interpreted languages.

The TURBO text editor is patterned after the widely used WORDSTAR word processing program. It is easy to learn and use, and provides a secondary skill which students can transfer to other word processing and editing situations.

TURBO compiles source code directly into native machine code. This means that TURBO PASCAL programs can be directly executed by the computer without having to have other programs present, such as an interpreter or a run-time module.

The compilation speed is extremely fast, usually on the order of seconds rather than minutes required by many other Pascal compilers. TURBO object program execution speed and code size compare very favorably with other Pascal implementations. Program size tends to be small and execution speeds fairly fast. I have found the TURBO compiler to be extremely reliable and almost completely bug free. All textbook and article Pascal programs that I have tried have compiled and executed without a hitch.

In my estimation, all of these features combine to make TURBO a more useful and well rounded for use in the computer classroom other Pascal implementations costing up to ten times more.

So - how is TURBO being used at Wasilla High?

I use TURBO as the Pascal for a two semester advanced placement computer science class. Students use it to write programs as assignments and I use it to write programs for computer assisted instruction. The course outline, which will be available at the conference, generally follows the guidelines for the advanced placement test in computers.

Pascal has also been ideal for developing programs for use by other classes. The lab at Wasilla High is available for part of the day for use by any teacher wishing to integrate the computer as a tool into the curriculum. This not only includes computer intensive classes, such as word processing skills, or accounting, but also general classes such as english or mathematics. Several of the Pascal programs currently being used are: a keyboarding skills tutor and drill, general mathematics drills and games (covering whole numbers, fractions, mixed-numbers, and decimal arithmetic), multiplication table drills, drills and games for factoring of quadratic equations, a reading skill and comprehension program, a data entry

program, an assembler language simulation program, an equation plotting program, and others. I have found that in-house development of these programs has considerable advantages over the purchasing of commercially developed programs. Although the quality of commerically developed programs appears to be improving, they often have limitations that severely restrict use. In house programs can be altered wherever needed to fit specific classroom or subject requirements and may be freely copied without worry concerning copyright laws or software protection devices used by manufacturers and distributors.

Is Pascal difficult to learn?

The Japanese play "Go", an ancient board game. Two players alternately place white or black stones at intersecting lines on the playing field. The rules are so simple that I could teach them to you in less than five minutes - and we could begin to play. But, it is said that you would then need a lifetime to master it. Pascal is like that in some ways. Really, it is a simple language; it is made up of a relatively small set of commands, rules and data structures. The basics are not difficult. The basics of Pascal can be fairly easy to learn if you don't approach the language with preconceptions derived from using BASIC, and if you don't expect to immediately match Pascal programs written by someone who has been using a procedural language for a long time.

I think Pascal is generally easier to learn and use than BASIC or LOGO. Don't hesitate to try it.

###

To order or inquire about TURBO PASCAL:

Borland International cost: $49.95
4113 Scotts Valley Drive
Scotts Valley, California 95066

Available at the conference:

- Course outline for high school level Computer Science course (advanced placement)
- Programs written in Pascal for use in the high school lab. They run on the IBM PC, IBM PC jr, and most IBM PC compatible computers. (We has Corona PC's at Wasilla High) These programs will be demonstrated, and you may copy most of them if you provide your own floppy disk.

Bibliography

Articles

Bromley, John P. Jr., "Pascal in High School," The Computing Teacher, Vol 11, No. 3, (October, 1983), 64-67. (Mr. Bromley uses Apple Pascal on Apple computers.)

Carol, David "The Versatile Turbo Pascal," Microsystems, Vol 5, Number 2 (February, 1984), 110-115.

Covington, Michael, "The Power of Turbo Pascal," PC Tech Journal, Vol. 3, No. 2, (February, 1985), 113-123. (This edition also contains other articles on Pascal.)

Woteki, Freiden, Levy, Bestul, and Stine, "Modular Implementations," Tech Journal, Vol 2, No. 6, (December, 1984), 155-174. (Although this is a comparison of 3 Modula-2 compilers it contains a benchmark comparision which includes Turbo Pascal. Modula-2 is an improvement on Pascal by N. Wirth.)

Books

Grogono, Peter. Programming in Pascal. Reading, Mass: Addison-Wesley Publishing Co., 1978.

Jensen, Kathleen and Wirth, Niklaus. Pascal User Manual and Report. Berlin: Springer-Verlag, 1974.

Schneider, G.M., Weingart, S.W., and Perlman, D.M. An Introduction to Programming and Problem Solving with Pascal. New York: John Wiley & Sons, 1978.

References

The Advanced Placement Program of the College Board. P.O. Box 977, Princeton, NJ 08544

LIBRARY MEDIA CENTER MANAGEMENT WITH MICROCOMPUTERS

Mary Claire Harris
Capital School Library
Juneau School District
Juneau, Alaska

Computers are receiving more and more attention from Library Media Specialists/Librarians as hardware and software prices continue to drop and applications for automating library routines become time/cost effective.

In this seminar, an analysis of when to automate and when not to automate will be discussed. Steps in systematic planning, which include analysis, development, and evaluation, will be presented along with an introduction to the general classes of software found useful in library management.

Sample applications and design templates will feature "Appleworks," an integrated database management system, word processor, and spreadsheet software package for the Apple IIe/c.

Templates and samples for a variety of reports follow the list of recommended reading.

RECOMMENDED READING:

Gillespie, John T., and Spirt, Diana L. Administering the School Library Media Center. New York: R.R. Bowker Company, 1983.

Naumer, Janet Noll. Media Center Management with an Apple II. Littleton, CO: Libraries Unlimited, Inc., 1984.

Truett, Carol. "To automate or not to automate: that is the question." School Learning Resources 4 (Winter 1985): 8-10.

TEMPLATES AND REPORT SAMPLES

Design Template for database file: BOOKORDER

```
Quantity:  (After ordering, used to indicate books received)
Author:
Title:
Illus:
Edition:
Publisher:
Year:
Price:
LC#:
ISBN:
Classification:
Review:
Note1:  (Flexible use of notes includes: direct order
Note2:  address, annotation after book received, name
Note3:  of person requesting, etc.)
Vendor/PO#: (Used only when generating an order)
```

Reports using file: BOOKORDER

Title of Report: BOOK ORDER
Field Sort Order: ISBN; Vendor
Fields Used: Quantity, ISBN, Title, Price, and a calculated field, TOTAL (Quantity * Price)
Print: Where Vendor contains ABS (Anywhere Book Supply); table type report

To: Anywhere Book Supply
City, State

Date: 2/3/85
PO #: 97345

Qua	ISBN	Title	Price	TOTAL
3	0-316-03131-3	World famous Murie	11.45	34.35
1	0-688-02604-4	Music, music for	10.51	10.51
1	0-8037-0066-0	Fox all week	8.89	8.89
2	0-87614-261-7	Tobias goes seal	7.95	15.90

Program totals, computes freight, etc. as needed.

Title of Report: NEW BOOK LIST
Field Sort Order: Author: Classification
Fields Used: Classification, Author, Title, Illustrator (if not blank), Publisher, Year, and Notes 1,2 & 3 used as the annotation
Print: All received; label type report

NEW ARRIVALS
IN THE LIBRARY/MEDIA CENTER

E MAR Marshall, Edward
Fox all week
illus by: James Marshall
Dial 1984
Fox returns in seven new adventures for beginning readers, one for each day of the week.

E WIL Williams, Vera B.
Music, music for everyone
Greenwillow 1984
Rosa, of "A Chair for My Mother" and "Something Special for Me", organizes her friends into the Oak Street Band and becomes a great success at a neighborhood party.

Title of Report: WILBUR
Field Sort Order: none needed
Fields Used: LC#: Classification
Print: All received with LC# not blank; table type report

Associate members of the Alaska Library Network/Washington Library Network (WLN) may report bibliographic holdings to WLN using WILBUR.

Files generated using this report form would be sent to WLN on a floppy disk or transferred via modem directly to WLN.

LC#	Classification
82-17186	E ALE
83-51441	629.13 ARD
83-51585	523.6 HAM
83-17070	629.8 BAL
83-50594	629.441 JAY
83-14196	E WIL
83-25273	574.5 CAT
83-20852	598 PAT

Design Template for database file: SOFTWARE

```
Classification: (Same scheme used for other materials)
Title/Version:
Producer/Date:
Equipment Required: (computer, memory, color, printer, etc.)
PO#/Vendor/$:
Building Location: (Permanent location other than library)
School Code: (Two letter codes used for each school)
Type: (tutorial, simulation, game, utility, etc.)
Subject1: (Broad: i.e. Language Arts)
Subject2: (Specific: i.e. Puncuation)
Grade Level:
Interest Level:
Special Uses: (i.e. gifted, speech, resource, etc.)
Modification:
Notes: (flexible use, usually short annotation)
Backup Policy:
```

Reports using file: SOFTWARE

Title of Report: DISTRICT SOFTWARE DIRECTORY
Field Sort Order: Title
Fields Used: Title/Version; Producer/Date; School Code
Print: All; table type report

DISTRICT SOFTWARE DIRECTORY
1985

School location code:

AB = Auke Bay E	GV = Glacier Valley E
CA = Capital E	HB = Harborview E
FD = Floyd Dryden MS	JD = Juneau/Douglas HS
GA = Gastineau E	MD = Marie Drake MS
MR = Mendenhall River E	

Title/Version	Producer/Date	School Code
Alaska Geography	Applecart, 1983	AB CA GA GV
Appleworks	Apple, 1984	AB CA FD GA
Card Prep	Library Software,1984	CA GV FD MD
Create Vocabulary	Hartley,1983	GV HB MR
Crossword Magic	L & S Computerware, 1981	AB CA GA GV
Gertrude's Puzzles	Learning Company, 1982	AB CA GA MR
Graphics Exhibitor	Koala Tech, 1984	JD MD MR
Human Body	BrainBank, 1981	GV FD JD MR
Report Card	Sensible Software, 1982	GA GV JD MR

Design Template for database file: SOFTWARE

```
Classification: (Same scheme used for other materials)
Title/Version:
Producer/Date:
Equipment Required: (computer, memory, color, printer, etc.)
PO#/Vendor/$:
Building Location: (Permanent location other than library)
School Code: (Two letter codes used for each school)
Type: (tutorial, simulation, game, utility, etc.)
Subject1: (Broad: i.e. Language Arts)
Subject2: (Specific: i.e. Puncuation)
Grade Level:
Interest Level:
Special Uses: (i.e. gifted, speech, resource, etc.)
Modification:
Notes: (flexible use, usually short annotation)
Backup Policy:
```

Reports using file: SOFTWARE

Title of Report: DISTRICT SOFTWARE DIRECTORY
Field Sort Order: Title
Fields Used: Title/Version; Producer/Date; School Code
Print: All; table type report

DISTRICT SOFTWARE DIRECTORY
1985

School location code:

AB = Auke Bay E	GV = Glacier Valley E
CA = Capital E	HB = Harborview E
FD = Floyd Dryden MS	JD = Juneau/Douglas HS
GA = Gastineau E	MD = Marie Drake MS
MR = Mendenhall River E	

Title/Version	Producer/Date	School Code
Alaska Geography	Applecart, 1983	AB CA GA GV
Appleworks	Apple, 1984	AB CA FD GA
Card Prep	Library Software,1984	CA GV FD MD
Create Vocabulary	Hartley,1983	GV HB MR
Crossword Magic	L & S Computerware, 1981	AB CA GA GV
Gertrude's Puzzles	Learning Company, 1982	AB CA GA MR
Graphics Exhibitor	Koala Tech, 1984	JD MD MR
Human Body	BrainBank, 1981	GV FD JD MR
Report Card	Sensible Software, 1982	GA GV JD MR

74 BEST COPY AVAILABLE

Title of Report: SCHOOL SOFTWARE LIST
Field Sort Order: Title (Can generate one by type, subject, or any combination to meet staff needs)
Fields Used: Classification; Title/Version; Producer/Date, Equipment Required; Type
Print: All; label type report

MICROCOMPUTER SOFTWARE

Alaska Geography
Applecart, 1983
Equipment required: Apple II+/e
Type: Tutorial 979.8/ALA

Card Prep
Library Software,1984
Equipment required: Apple IIe, printer
Type: Management 025.7/CAR

Create Vocabulary
Hartley,1983
Equipment required: Apple IIe
Type: Drill/practice 428/CRE

Crossword Magic
L & S Computerware, 1981
Equipment required: Apple II+/e;printer
Type: Utility 793.73/CRO

Gertrude's Puzzles
Learning Company, 1982
Equipment required: Apple II+, color monitor
Type: Problem Solving 153.41/GER

Graphics Exhibitor
Koala Tech, 1984
Equipment required: Apple IIe/Koala Pad
Type: Utility 001.64/GRA

Human Body
BrainBank, 1981
Equipment required: Apple IIe
Type: Tutorial 611/GAR

Report Card
Sensible Software, 1982
Equipment required: Apple IIe
Type: Utility 371.1/REP

Design Template for spreadsheet file: LEDGER

	A	B	C	D	E	F	G	H
1								
2		Label Descriptions:						
3			A1 Budget					
4			B1 Date					
5			C1 Purchase Order Number					
6			D1-E1 Vendor					
			F1 Encumber (Amount of Purchase Order)					
			G1 Expended (Actual amount paid - includes freight, etc. - expended may change until order is complete)					
			H1 Encumbered Balance: Formula @MIN(Fx-Gx)					

Totals: Formulas @SUM(Fx.Fx); @SUM(Gx.Gx); @SUM(Hx.Hx)

Sub/Balance: Formula @MIN Ax - (Gx+Hx) (Here 'x' indicates the row of totals immediately preceeding each budget category.)

Report for Budget Category: 451 Teaching Supplies

Budget	Date	PO#	Vendor	Enc	Exp	Enc Bal
1400	(451 Teaching Supplies)					
	Jul 1	20147	Comp. 1	30.75	29.48	1.27
	Jul 1	20148	Comp. 2	49.7	50.51	-.81
	Jul 1	20150	Comp. 3	356.85	298	58.85
	Jul 1	20151	Comp. 4	32.4	28.6	complete
	Jul 1	20152	Comp. 5	565.5	499.95	65.55
			Totals	954.75	826.55	124.4
			551 sub/balance			449.05

Trends and Issues in Logo: Implications for Teachers

Mary Susan Henrikson
Computer Education Specialist
Northwest Regional Educational Laboratory
Portland, Oregon

When Logo was first developed for large computer systems, it lacked the graphics capabilities many now associate with it. Activities in the early versions of Logo utilized list processing. Years of research and development at M.I.T. produced the turtle and "turtle" graphics. In addition, the evolution of the microcomputer brought the Logo language out of the research laboratory and into classrooms and homes. This session will be involved with discussing trends, issues, and new products relating to the Logo language and its use in the classroom.

The first concern is that of revisions, implementations, and new versions of the Logo language. Almost all microcomputers (Apple, Acorn, Atari, Coleco, Commodore, IBM, Tandy, etc.) at this time have some version of Logo available. Some computers, such as the Apple and IBM, have several versions. This presentation will deal with new Logo products for Apple products.

The latest versions for the Apple II line include Apple Logo II and Sprite Logo for the Apple. Both Apple Logo II and Sprite Logo were developed by Logo Computer Systems, Inc (LCSI). Apple Logo II is an enhanced version of the original Apple Logo. It runs on either the Apple IIe with 128K of memory and one disk drive or the Apple IIc. A color monitor is not essential but recommended. Features of Apple Logo II include:

- Enhanced list processing
- Dribble files

- Built-in print functions (screen dumps)
- Music procedures
- Pro-DOS
- Fill command
- More memory
- Improved editing features

Apple Logo has few new graphic functions. Colors available include typical high resolution colors: black, white, orange, blue, green, and violet. Sprites, while not essential for Logo, are nevertheless a nice feature and are not included with this version. While the system is still relatively slow, the ability to use Pro-DOS does increase the efficiency of execution.

Sprite Logo for the Apple runs on the Apple IIe, Apple IIc, or Apple II+ with 64K of memory and one disk drive. This version requires a sprite board (included with the software) and a monitor. A color monitor is not absolutely necessary, but without one the color capabilities of the system would be wasted. Features of Sprite Logo for the Apple include:

- Apple Logo command compatibility or semi-compatibility
- 32 built-in sprites
- Sprite (graphics) editor
- Multiple turtles (30)
- 16 colors
- Speed, line, and collision detection functions

The Sprite Logo program diskette is not copy protected but will not run without the sprite board. There have been reports that the sprite board sometimes causes overheating, even with an Apple IIe but a fan usually alleviates the problem. The sprite board can be a problem in that the monitor must be plugged into it. Printing pictures may also be a problem although long time Logo users are probably used to that by now.

There is one version of Logo currently available for the Macintosh, Expert Logo. Expert Logo can run on a 128K or 512K Macintosh. Features of Expert Logo are:

- Apple Logo command compatibility or semi-compatibility
- Uses Mouse
- Pull-down Menus
- Customized Menus
- Up to 100 times faster
- Compiled instead of Interpreted
- Uses Bunny Graphics for 3-D
  - Pitch, roll, and yaw
  - Bunnies can live in cubes and spheres
- Array functions

Expert Logo is probably more appropriate for high school or college use. The advanced list processing and the change to a compiled language may make Logo appropriate for application programmers. Commands are included in pull-down menus for easy programming. Debugging is similar to other Logo versions. One drawback of Logo on the Macintosh is the inability to utilize color monitors at this time. Logo for the Macintosh, developed by LSCI, will be available in early spring.

The second part of this session is devoted to trends in the use of Logo. A recent survey of subscribers by the National Logo Exchange found that approximately 52% of Logo use is concentrated in the upper elementary grades 3-6. The percentage of Logo use declines significantly after sixth grade, falling to 14.01 in grades 7-9 and a meager 2.21 in grades 10-12. The most popular version of Logo is Apple Logo at 40%, followed closely by Terrapin Logo at 31.72% and Krell Logo at 8.97%. Not surprising is the dominance of use of the Apple II+ or IIe as a mode of Logo instruction in the classroom at 80.69%.

If you have attended Logo sessions such as Logo 84 at M.I.T., you have probably noticed two distinct groups have seemed to emerge regarding the philosophy and method in the teaching of Logo. One group typically supports a programming, mathematical theme. The other upholds Papert's (1980) concept of a Piagetian approach and the development of powerful ideas. The two groups appear to be at opposite ends and at some sessions and conferences heated words have been exchanged. Uri Leron in his article, Logo Today: Vision and Reality, proposes that some mixture of the two as the ideal method of involving a child with Logo. Leron supports a "quasi-Piagetian learning" (QPL), which requires the teacher to take an active role but does not inhibit the creativity of students. It is his opinion that while most children may master the syntax of the language, few will advance beyond the classification of "hacker" without some instruction.

In my experience, the utopian teacher, one that promotes an environment of "powerful ideas," should have four characteristics. That teacher should be

- Comfortable with children
- Comfortable with computers
- Comfortable with Logo
- Knowledgeable in the philosophy of the Logo language

Too often, teachers possess too few of the above requirements. In Leron's (1985, p. 29) view, "It is possible that the more prevalent problem in the practice of Logo will not be too little learning..., but rather too much teaching--and of the wrong sort." There are unfortunately a limited number of teachers that have the training and characteristics necessary to implement Papert's theories into practice.

Extensive research pertaining to Logo is usually qualitative and contains detailed descriptions of children working within a Logo environment. The bulk of research conducted at M.I.T. is rich with descriptions and analysis of childrens' activities. While supporters of "Piagetian" Logo, are supportive of this type of descriptive research, skeptics suggest that Logo will only be accepted when substantial research is presented that confirms it as a means for increasing problem-solving, math, or reflective skills and thus influencing school achievement. Results, to date, from quantitative experiments attempting to measure this type of Logo influence have typically yielded disappointing blows to Logo supporters. Leron contends that evaluative methods for determining the success of Logo are largely unavailable and irrelevant at this time. He maintains that such evaluation is inappropriate until a method involved with the teaching of powerful ideas rather than facts is implemented in the classroom. Unfortunately the "back-to-basics" movement coupled with negative research results, may dictate the omission of Logo from the classroom without truly learning the potential of this powerful learning and thinking tool.

Logo References and Resources

Books

Abelson, H., and diSessa, D. Turtle Geometry: The Computer as a Medium for Exploring Mathematics, MIT Press, 1981

Abelson, H. Logo for the Apple II, BYTE/McGraw-Hill, 1981.

Papert, S. Mindstorms: Children, Computers and Powerful Ideas, Basic Books, 1980.

Watt, D. Learning with Logo, BYTE/McGraw-Hill, 1983.

Magazine Articles and/or Issues

Brackney, L. "TRS Turtle," 80 Micro, No. 37, Feb. 1983, p. 116-120.

Byte Magazine, (Special Logo Issue), Vol. 7, No. 8, August 1982.

Carter, R. "Try it Turtle Topics: Drawing Polygons," Classroom Computer News, Vol. 3, No. 5, April 1983.

Classroom Computer Learning, (Special Logo Issue), Vol. 4, No. 3, October 1983.

Clements, D. "Logo Programming: Can it Change How Children Think," Electronic Learning, Vol. 4, No. 4, January 1985, p. 28, 74-75.

Clements, D. "Print Materials on Learning Logo for Children and Their Teachers," Educational Technology, Vol. 24, No. 1, January 1984, p. 53-56.

Hoffer, B., and Semmes, P. "Computer Haiku: A Classroom Experiment with Logo in the Language Arts," The Computing Teacher, Vol. 12, No. 4, Dec/Jan 1984-85, p. 52-53.

Kramer, S. "Word Processing in a Logo Environment," Electronic Learning, Vol. 3, No. 6, March 1984, p. 70.

Leron, U. "Logo Today: Vision and Reality," The Computing Teacher, Vol. 12, No. 5, February 1985, p. 26-32.

Lough, T. and McCurdy, K. "Gradations of Logo," The Computing Teacher, Vol. 12, No. 4, Dec/Jan 1984-85, p. 56.

Mace, S. "Atari Logo," Infoworld, Vol. 6, No. 4, January 23, 1984, p. 57-60.

Maddux, C. "Will the Potential be Realized?," Educational Computing, Vol. 4, No. 1, January 1984, p. 76-77.

Renne, M. "IBM Logo," Infoworld, Vol. 6, No. 24, January 11, 1984, p. 69-70.

The Computing Teacher, (Special Logo Issue), November 1982.

The National Logo Exchange Research Directory, R. Spona (Ed.), The National Logo Exchange, June 1984.

Thornburg, D. "Early Grade Ideas," Classroom Computer Learning, Vol. 4, No. 6, March 1984, p. 70.

Wappler, R., and Lough, T. "Logo Grows Up," Electronic Learning, Vol. 4, No. 4, January 1985, p. 38-42.

YPLA Software Exchange, The Computing Teacher, Vol. 11, No. 7, March 1984, p. 50-57.

User's Groups and Newsletters

Boston Computer Society Logo Users Group, One Center Plaza, Boston, MA 02108.

Friends of LISP/LOGO and Kids, 436 Arballo Dr., San Francisco, CA 94132.

Friends of the Turtle, Box 1317, Los Altos, CA 94022.

Logo Times, 99'er Magazine, Box 5537, Eugene, OR 97405.

Logophile, Logo Special Interest Group, c/o Higginson, Faculty of Education, Queens University, London, Ontario, K7L 3N6 Canada.

Manadnock Area Logo User's Group, c/o Dan and Molly Watt, Greg Lack Rd., Antrim, NH 03440.

Microquests, Martin-Bearden, Inc., P.O. Box 337, Grapevine, TX 75051

National Logo Exchange, Box 5341, Charlottesville, VA 22095.

Young People's Logo Association, 1208 Hillsdale Dr., Richardson, TX 65081.

Michael Hoar and Ken Klawunder

Computer Assisted Curriculum Development and Resource Management

The presentation will describe the methods computers are used by Alaska Gateway School District to assist curriculum building, use and updating.

The philosophy of the district concerning computer use and curriculum will be discussed first to set the stage for the development process. Outcome Based School, using the Mastery approach, are predicated on the teaching and learning of predetermined outcomes. The entire learning process is based on the mastery of these outcomes. Then "how" can we assure that the curriculum has staff buy-in, will be used constantly in lesson design, and will include all the resources and activities available in the district to teach a particular outcome.

Computerization of the curriculum was necessary to accomplish all the necessary functions.

Following is a brief outline of the topics to be covered.

1. Alaska Gateway School District Curriculum Development Process.
   - A. Curriculum Development.
     1. Curriculum Meetings
     2. Parts of the Curriculum Document.
   - B. Use of the "Lisa" in Curriculum.
     1. Outcomes stored and generated.
   - C. Teacher Use of Curriculm Document
     1. As a design for lessons
     2. Updating
        a) Outcomes
        b) Resources
        c) Activities
        d) Criterion Referenced Tests
   - D. Curriculum Transfer to IBM Mini 3600 and School Access.
     1. Re-typing
     2. Continuous up-dating
     3. School sight generation

II. Resource Center Materials Database.

A. 60 meg of Corvus hard disk.

1. One 20 meg. Const. I network for Computer Cat. Level 5.

2. One 40 meg. Const. II network for the District Resource Center.

a) the Const. II network has the operating systems to drive the Corvus Concept Computer

b) We also have D.O.S. 3.3, P-System and CP/M operating on this network.

c) An I.B.M.- PC will be added to the network by May.

B. Database selection and design for cataloging resource materials within the district.

1. PD Base

2. ADI Aladin

C. Methods of information collection from throughout the district.

D. Districtwide electronic mail system for making resource center materials requests. HELP WANTED!!

Dialog Workshop

Using A Dialup Database in the Classroom

Presented by Daniel L. Hoffman

One of the major problems facing secondary, and even primary, students in Alaska is a derth of reference sources from which they can get any meaningful quantity of material for research reports. This affliction plagues teachers as well. I'd like to say that the solution to all of our problems is at hand, but I can't. What I can say is that the answer to some of our problems is here. But as in all good science, one answer often generates its own set of problems.

First, let us define the problem. You have a student who is need of some reference material on an assigned topic for his/her History, Science, Civics or English class and he/she has exhausted all of the local library's selection of encyclopedias and magazines. Being left somewhat short of the required pages for the report, the student begins to do the only sensible thing left to him/her under the circumstances, he/she begins to write larger, double spaces on the typewriter or just plain begins to make up facts. But what if this person had access to a virtually unlimited source of information with which to pursue the assigned topic? Might you not expect to receive a little better paper? And might not the student get a little more out of all the work he/she had to put into the paper in the first place? The fact of the matter is that this is exactly what our goal in education should be, to enhance the intrinsic rewards of the educational process by making the gain in knowledge worth the effort the student has to make. This is the wage the student earns in his/her work while he/she is in school.

Alvin Toffler wrote a book which I am sure most of you are familiar with. It is titled The Third Wave. In it, the author speaks about the age in which we now find ourselves. It is an age of information and of information handling and don't our students deserve to be a part of this age as much as we. In fact, it is their right as

they are the inheritors of this age. So how can we provide them with the opportunity to take full advantage of this information age of ours and theirs.

Addressing a seeming answer to our current crisis, I wish to talk to you about Dialog, a database system that contains more than 170 different sources of information for the inquisitive mind. All one needs to access this vast source of information are some simple hardware devices such as a computer, a printer, a modem and a telephone, and, if possible, a software package that will allow you to use all of the aforementioned items. The software isn't really necessary, but it makes it so much easier.

Dialog began in 1981 as an outgrowth of the Lockheed Corporation's information retrieval service of the same name. It offers more than 70 million abstracts in its more than 170 on-line databases. Information from the databases includes references to journal articles, magazine articles, newspaper articles, conference papers, technical reports, and patents. The information available covers all areas of science, technology, business, engineering, medicine, social science, current affairs, news and humanities.

The following catagories can be researched through the use of the Dialog databases:

- Agriculture/food science
- Business/economics
- Chemistry
- Contracts/grants/current research
- Education
- Energy/environment
- Government publications
- Law
- Materials science
- Medicine/biosciences
- News/ current affairs
- Patents
- Science
- Social Science/humanities
- Technology/engineering

Dialog has also recently added a service for personal computer users called Knowledge Index.

Dialog services are available 22 hours a day. Here in Alaska it is not available only between the hours of 6 and 8 P.M. local time, Monday through Thursdays; Additional available hours are midnight to 5 P.M. on Fridays; 5 A.M. to 5 P.M. on Saturdays; and 9 P.M. to midnight on Sunday. Friday, Saturday and Sunday times are Pacific time.

Dialog also offers a Customer Service department which offers an 11-hour-a-day customer assistance service.

Customers pay for the Dialog service on an as-used basis; the charges are based only for the searches and number of off-line prints requested. The average search runs approximately $5 to $15[1], but my experience of classroom searchs by students runs somewhere between $2 to $5.

So what can this service do for you? Through simple, basic commands, Dialog can provide a powerful research tool to both you and your students. As the databases are continually updated, they provide the most current information possible. Subjects can be selected by keyword, person name, place name, date, language, geographic region, organization, and many other elements. Full texts of records can be searched

for specific information. Once a search is complete, you can order the full text of the documents located through the search via Dialorder, an on-line document ordering service. Even if you are not certain which database will be the most helpful to you, a Dialindex facility is there to assist you.

We've been using Dialog in a pilot program in the Matanuska-Susitna Borrough since October. Teachers from ten schools were given a one-day in-service on how to conduct Dialog searches, $100 and a password to use to practice using the service. We met again in February to compare notes and see if the service was worth the expenditure. These are some of the comments that came back from the users:

"Very definite value here as a professional growth tool."

"Incredibly fast access to references at a modest cost. This is an idea whose time has come."

"It should definitely be a part of the computer curriculum at the secondary level."

"I think it is a nice program because it is easy for me to get information for my classes and projects."

"Please continue this program. In the brief time I have had to spend with this system, its value to me professionally and to the school is limitless. Please keep it on line!"

"It's helped in my presentation for my gymnastics unit as well as for my upcoming track and field unit."

"It's nice to be able to get a brief overview of a book or article. You don't have to waste time skimming through, you can get right to the heart of the matter. In the time it saves in research, it's more than worth it."

"Students should be exposed to the program at least one time."

"Unlimited uses in History. Excellent idea."

"This is what I've been waiting for! I've wanted something that lets me know what is happening in the rest of the educational world. I love to research and enrich classroom practices."

"Dialog cuts down on the time necessary to find material for an area of study. This not only saves time but also encourages me to look for more information about my topic then I may have in the past."

These comments come from students, teachers, counselors and administrators within the District. Of the thirty-three information retrieval services, which included

services such as Compuserve and The Source, surveyed and rated by the Datapro Research Corporation, Dialog rated in the top three for Technical/Instructional Support, Documentation (first), Ease of Use and Overall Satisfaction.

In my own classroom, 7th and 8th grade gifted students, we have conducted searches which have ranged from research on wolf diets to the future use of computer technology. The students find it exciting to be able to communicate with another place in the world, to be able to get information at a very rapid rate, to be able to get information that is normally not available to them and just to be able to use an "adult" or "teacher's" tool. And as stated above by a district teacher, the students look for more than they normally would.

Using Dialog has had an impact on the development of my, as well as others, students in the following ways; because the students have to organize a search rather thoroughly, they have to ask questions about their topics. Key words are generated, large general topics are narrowed to specific areas of interest. Additional areas of interest are discovered and explored. And the kids vie with one another to get to use the computer to get their information.

Which brings me back to my original contention, our main job is to provide the student with enough positive feedback for his/her endeavors and efforts to make all of the work worth the knowledge gained. Just the adventure of being able to use a modern contrivance as telecommunications is enough to get some started. And once started, its easier to keep the ball rolling.

[1] Datapro Complete Guide to Dial-up Databases, Datapro Research Corporation, Delran, NJ, 1984.

Presentation of Dialog in Class Pilot Project

The presentation to the AACED will include the above comments as well as

examples of searches, a brief look at available hardware and software for telecommunications and a brief description of actual pre-search classwork and log-on procedures.

COPYRIGHT AND COPYING RIGHTS AND RESPONSIBILITIES

Norman D. Holthouse, Ph.D.
Director of Management Information Systems
Anchorage School District
Anchorage, Alaska

The copyright and copying controversy remains a complex and emotionally charged issue among the microcomputer user community. While not unique to APPLE microcomputers and APPLE based software, this paper will focus on the problem in the APPLE domain because of the wide spread use of APPLE microcomputers and related software in school districts throughout Alaska. Most users are by now painfully aware of the problems associated with two conflicting requirements: (1) the need to respect copyright laws and protect the rights of software authors to insure that they are not deprived of royalties that are lost by illegal distribution of pirated software, (2) the need to maintain backup copies of programs and data files to protect against the destruction or failure of the diskettes on which they are stored. Compounding the issue has been the emergence of two related phenomena; the use of software protection schemes designed to make software diskettes uncopyable by normal copy routines, and the commercial development of sophisticated copy programs designed to copy the diskettes that have information stored on them in a protected format.

The concept of backup is one that is often painfully learned by new users of microcomputers. Backup or archival copies of important data and software has always been recognized as professional practice in data processing centers. The recognition and experience of magnetic media failures led to the practice of regularly scheduled backup of all files very early in the history of computing. A data processing manager who failed to implement backup

procedures would be quickly dismissed. The same need for adequate backup is felt and required by the microcomputer user as well. The fact is that diskettes do become defective with the accompanying loss of the information stored upon them. When failures occur, the careful user immediately makes a duplicate copy from the backup that was carefully stored for such purposes. Unfortunately, the software protection schemes have prevented this normal backup procedure by many honest users.

There is, unfortunately, another side to this issue. The protection schemes utilized by the software companies were not designed to extract extra profit from the users, but arose from a realization by software companies that they were being deprived of sales by the practice of software piracy. Software piracy may take many forms, but is usually the distribution of duplicate copies of software protected by copyright without the consent of the copyright holder. The extent of this illegal practice is not known precisely but it is estimated that for every commercial sale of a software diskette there are at least nine pirated copies in existence. The copy protection schemes were instigated in an attempt to thwart the software pirate. The software users should also recognize the fact that copying abuse could eventually lead to either a lack of new development in needed software, or increased prices necessary to compensate the author for missed royalties. Good quality software requires hundreds of hours of programming effort for which the professional author has every right to expect a just return.

It is apparent that there are two sides to this complex issue, with many shades of gray separating them. It is probably just as important to develop a proper social perspective among students on these issues as it is to provide them with understanding and skills involving computers.

What are our responsibilities in this area? Each district should state in writing the intent to adhere to the provisions of copyright laws as they apply to computer software. Recognizing that there continues to be several areas that are open to differing legal interpretation, the following procedures are recommended as practical guidelines:

* Illegal copies of copyrighted programs may not be made or used on District owned hardware.
* The ethical and practical problems resulting from software piracy should be considered an important aspect of computer literacy.
* Faculty and students who utilize District hardware to make multiple copies of software protected by copyright for other than legitimate backup purposes should be informed that such practices are not allowed and that future violations will result in loss of use privileges.
* The legal or insurance protection of the District will not be extended to employees who violate copyright laws.
* The principal of each school site is responsible for establishing practices which will enforce the District policies at the school building level.

Accompanying these responsibilities are certain rights which should also be communicated and understood:

* The purchaser of software protected by copyright has the right to create and possess a backup copy of the original. Multiple archive copies may be appropriate in those instances where different sets of data are also created on a diskette.

* The fact that a software diskette is "copy protected" does not in any way negate the right to make legitimate archive copies.
* The possession and use of special copy programs designed to copy "protected diskettes" is a legal right as well as sound professional practice.
* A licensing agreement which allows a District unrestricted use and distribution of a software package should be viewed as a highly desirable arrangement. The educational community should attempt to influence vendors to make such licensing agreements widely available.

In this rapidly changing world of technology, these rights and responsibilities are set forth as guidelines that attempt to address a complex issue in a rational manner. There are circumstances and conditions for which these guidelines do not seem to fit. Conditions such as the following may not be clearly addressed by the current copyright laws:

* What constitutes "fair educational use" as applied to educational software?
* Can a software program be "booted up" and used in several machines at the same time?
* Can a software program be installed on a time-sharing or disk-sharing network?
* At what point do extensive modifications to an original program result in a program not protected by the original copyright?
* Can algorithms or conceptual techniques be borrowed from one program and used in a new creative work?

The existence of such gray areas should not be taken as license to ignore the intent of copyright protection, but should instead be viewed as part of the growing pains that accompany the rapid growth and development of computer technology.

A review and demonstration of software and firmware that are designed to facilitate the creation of a backup copy of "copy protected" diskettes will be made during the presentation on Copyright and Copying. Handouts addressing the various products available including parameter lists will be available at the session. For the reader unable to attend that session the following are recommended as state of the art copying aids for software designed for the APPLE II family of microcomputers.

Essential Data Duplicator II
Utilico Microware
3377 Solano Ave.
Suite #352
Napa, CA 94558

Wildcard 2
Central Point Software, Inc.
9700 SW Capitol Hwy., #100
Portland, OR 97219

NDH/mr

THE RESULTS OF AN EXPERIMENT WITH A COMPUTER IN EACH CLASSROOM

Norman Holthouse
Director, Management Information Systems
Anchorage School District
Anchorage, Alaska

Ralph Van Dusseldorp
Professor of Education
University of Alaska
Anchorage, Alaska

1. BACKGROUND

In the Fall of 1983 the Anchorage School District began a pilot project that involved placing an Apple microcomputer in each classroom of two elementary schools. The objectives for the pilot project are to:

1. Improve student achievement in spelling and problem solving in mathematics.
2. Develop positive attitudes of teachers, students, and parents towards computers as a teaching/learning tool.
3. Show that teachers who have computers in their classrooms are more involved in managing computer learning activities for their students than those who send children to a computer lab.
4. Show that when a computer is available on a daily basis for use as both a teaching and a learning tool there is a greater integration of computer activities into the curriculum.
5. Show that children who are routinely exposed to a computer as a piece of classroom equipment for information retrieval and learning develop more positive attitudes towards computer than those who see the computer as a special machine reserved for extraordinary tasks or situations.

Huffman and Willow Crest were chosen to be the pilot schools for the project. Each of those schools has about 18 classrooms and about 450 students. A computer was placed in each classroom of those two schools. It was decided that Huffman would use computers primarily to teach arithmetic problem solving and

Willow Crest would use theirs primarily to teach spelling. The District also provided computer software and inservice training for teachers.

Two schools that had computers located in a central computer laboratory rather than in individual classrooms were chosen to serve as comparison schools. They were Birchwood and Mountain View.

Evaluation of the project was based on attitude questionnaires filled out by students, teachers and parents in November 1983 and may 1984; Iowa tests of Basic Skills results from April 1973 and April 1984; interviews with students, teachers, principals and project administrators; examination of records and computer software; and classroom observations.

2. RESULTS

Student Use of Computers

Project guidelines called for each student to use the computer 30 minutes a week. It was originally anticipated that students would use computers independently and that each student would use the computer for three ten minute blocks of time a week. It was discovered that 10 minutes was too short a time so most students ended up using the computer twice a week for 15 minutes each time. Also, in most cases students used the computers in pairs rather than individually. That seemed to work better than individual use and was easier to schedule and manage. About two-thirds of the teachers had their students use the computer in pairs rather than individually.

Teacher Management of Student Computer Use

The scheduling and management of student computer use was a difficult problem for teachers early in the project. There was a logistics problem in seeing that each student used the computer two or three times a week at times the students were not engaged in other activities. After the first few weeks most of the teachers had this problem nicely resolved. In most cases the time at which students used the computer was quite flexible. Some teachers established an order in which students would use the computer while others left the order up to students. When students completed their turn they simply checked themselves off and notified the next students that it was their turn. At the end of the year no teachers felt they were unsuccessful in scheduling and managing their students' use of the computer.

Another problem in management was in making sure that the software used by students at a particular time matched the curriculum and the needs of the particular students. Teachers were not completely successful in doing that, partly because there frequently was not appropriate software available to meet particular needs. At the end of the year about one-third of the teachers felt they were unsuccessful in matching the students' use of computer to their learning needs and in matching the students' use of the computer with the regular teaching activities.

Student, Teacher and Parent Attitudes

Attitudes of pilot and comparison school students, teachers and parents toward the use of computers for instruction in schools were measured at the beginning of the project and at the end of the year. Attitudes of all three groups - students, teachers and parents - were extremely positive, both at the beginning of the project and at the end of the year. It would be difficult to imagine any other educational innovation being popular with all three groups.

Student Attitudes

Student attitudes were very positive both at the beginning of the project and at the end of the year. There was also very little change in student attitude during the course of the year. That may have been because student attitude was already so positive at the beginning there was very little room for further gain. The results for all four schools combined showed that:

* By the end of the year almost all students (99.5%) had used a computer in school.
* At the beginning of the project 57 percent of the students had used a computer at home or someplace else other than at school.
* At the end of the year about one-third of the students had computers in their homes that they could use.
* Almost all (94.8%) felt computers are important to have in school.
* Almost all (97.3%) felt that computers can help them learn.
* Almost all (92.6%) felt they would like to use computers more in school.
* Almost all (97.8%) felt that computers are fun.

A comparison of student attitudes between the pilot and comparison schools and among all four schools revealed very little differences.

Comparison of student response by grade level also showed very little difference between grade levels.

Teacher Attitudes

The attitudes of teachers in both the pilot and comparison schools toward computers were very positive. The results showed that at the end of the year:

* All teachers agreed that the computer can be an important learning tool.
* About two-thirds of the teachers had purchased or were seriously considering purchasing a computer of their own.
* The majority of teachers feel that the best place for students to use a computer is in the classroom rather than a central location.
* About two-thirds felt they could use more than one computer in their classroom.
* About half felt a computer can help manage their classrooms more effectively. One-third expressed no opinion on that item.
* Most felt that computers can be effectively used in ways other than drill and practice.
* Very few of the teachers indicated that they were still a bit afraid of computers.

A comparison between pilot school and comparison school teacher attitudes showed that the pilot school teachers were just a little more positive toward computers than the comparison school teachers.

A comparison by grade level showed that first grade teachers were a bit less positive toward computers than were teachers at other grades.

A comparison between teacher responses at the beginning of the project and the end of the year showed a slight gain but teachers were already so positive at the beginning there was little room for improvement. The biggest change was in the item that showed that fewer teachers had a fear of computers at the end of the year.

Parent Attitudes

Parent attitudes were very positive, both at the beginning and at the end of the year. At the end of the year:

* 95% felt that it is important for schools to have computers.
* 96% felt that a computer can be an important learning tool.
* 62% had purchased or were seriously considering purchasing a computer.
* 66% felt their children should have more time on a computer than was available during the project.
* There was little difference in parent attitudes among the schools but the pilot school parents were slightly more positive than the comparison schools parents.
* There was little difference in attitudes between grade levels of children.

A comparison between attitudes at the beginning of the project and the end of the year showed a small gain over the year. However, parent attitudes were already so positive at the beginning of the project that there was little room for improvement.

An analysis of comments made by parents showed that many stressed the computer literacy aspect rather than using computers as teaching tools. Perhaps that is best exemplified by a parent who commented, "I don't know how computers help kids learn, but I know its important for everyone to learn to use computers."

Student Achievement

The results of the Iowa Tests of Basic Skills (ITBS) administered in April 1983 and April 1984 were used to determine gains in spelling and mathematics achievement. Only students who took the ITBS both years were compared.

Spelling Achievement

One of the pilot schools, Willow Crest used computers primarily to teach spelling. Spelling achievement gains for Willow Crest students were greater than would be expected from national norms and were greater than the other pilot school or the comparison schools. Thus it appears that computers were effective in increasing spelling achievement.

Mathematics Achievement

Huffman was the pilot school that used computers teach mathematics, primarily mathematics problem solving. Comparison with national norms showed that Huffman students gained almost exactly what would be expected in one year. Comparison with the other pilot school and the two comparison schools did not show Huffman gaining more in mathematics problem solving than those other three schools. Thus, it does not appear that the use of computers was effective in raising achievement in mathematics problem solving.

Teacher End of Year Information

In addition to the computer attitude questionnaires, at the end of the year pilot school teachers reported pilot project information through another questionnaire. The results were:

* Most teachers felt that the use of computers increased their students' achievement in spelling and arithmetic problem solving.
* Most teachers felt that they could effectively use two or three computers in their classrooms.
* Almost all teachers felt they would rather have computers in their own classrooms rather than in a central location.
* Most computers were in use about half of the time.
* Most teachers felt that computers were effective for students of all ability levels.
* Teachers indicated that if they had as much hardware and software as they needed the average student could effectively use the computer about 45 minutes a week.
* An indication of teachers' interest in improving their use of computers is that almost all said that, if allowed, they would take a computer home during the summer in order to better plan and learn to use it.

Presentation Abstract

for

"Spread Sheet and Data Base Software"

by Mike Johnson, Ph.D.
Assistant Superintendent
Copper River School District

Objectives: *Participants will:*

a. State generalizations about the differences between spread sheet and data base software.
b. Identify where they might use spread sheet and data base software in their jobs.
c. State the differences between a *record* and a *field*.
d. State generalizations about the practical definition of what a *DATA BASE* is and how different data bases may be useful to them as administrators.
e. List what they see as exciting about the use of computers and management information in school administration.

Anticipatory Set: "By the end of this presentation I plan to show you how you can *protect your tale* and keep ahead of the lynch mob and at the same time add an interesting and intellectually stimulating aspect to your job and a new hobby to your lifestyle."

Input & Modeling: The presenter will provide information on transparencies and by lecture about the differences and uses of Spread Sheets and Data Base Software; including actual spread sheet and *DBASE II*® (Ashton-Tate) programming written and used by the presenter for various management applications.

Checking Mark *R* for *Record* and *F* for *Field* on the spread

Checking Understanding & Guided Practice:

Mark *R* for *Record* and *F* for *Field* on the spread sheet and data base examples passed out in class. List three differences between spread sheet and data base software and list three possible uses for each in your job. Define what a data base is and how it might be used in your job. Make a list of possible data bases that you use on your job on a daily basis and identify which ones use the same base information. Define the fields for a small data base that you could use in your job. Define the length (number of characters) that you would need in each of those fields. Define a *Key Field*. Identify the *Key* field in your data base. Make a guess as to the possible number of records you may have in your data base. Define what a *REPLACE* statement is in *dbase II #* programming.

INDEPENDENT PRACTICE:

Write a simple equation for a *REPLACE* statement. Define what *SUM* equation is in *CALC* programming. Set up a simple *CALC* spreadsheet. Write a simple *SUM* equation for your *CALC* spreadsheet. Find the *WHAT IF* statement in the *CALC* spreadsheet passed out in class. List what you see is exciting about these software programs and how they can help you in your daily job.

Spread Sheet and Data Base Software
Mike Johnson, Presenter
ARCED Convention □...□ April 4-6, 1985

Some of the differences between Spread Sheet and Data Base software are as follows:

- **Spread sheets** are not as flexible as **data bases**.
- What you type on the screen is what you get when you print out a **spread sheet**.
- A **data base** can allow you to design input screens that a clerical worker can use to input information.
- A **data base** can allow you to create a variety of output formats and reports.
- A **data base** allows you manipulate information and create new information in ways that spread sheets cannot match.
- **Summary**: **Data base** software gives you more flexibility at all phases - data input, manipulation, and output.

Uses of Spread Sheets and Data Bases:

- Spread sheets function best for fixed table data, for example, salary schedule information and simple

accounting spreadsheets.

- Data bases work best for more complex information storage and data manipulation, for example, student attendance and other records.

Definitions:

Both data bases and spread sheets operate on a **grid matrix** where there is a horizontal and verticle axis.

A **field** is a specific piece of data or information that generally make up the verticle axis or colums in a data base.

A **record** is a specific collection of information based on a **key field** and generally makes up the horizontal axis on the grid matrix.

A **key field** is completely individual that differenciates each individual record in data base from every other record.

Each **field** is limited by the **number of characters** that are possible in each field. The programmer must determine the maximum number of characters possible in each field when setting up the data base.

A **replace command** statement fills that **field** in each record of the data base by one command and the programmer does not have to type the information into each field for each record.

A **Sum equation** or **Sum command** automatically adds individual fields (columns) or groups of fields.

Spread sheets and data bases allow the programmer to forcast information and trends in data. These are known as ***What If*** statements or spreadsheet designs.

YRS. EXP.	# TEACH	BA+0	total	NEW BA+0	NEW TOTAL
0	2	28322	56644	29667	59335
1	0	30872	0	32338	0
2	0	33419	0	35006	0
3	1	34552	34552	36193	36193
4	1	35685	35685	37380	37380
5	1	36817	36817	38566	38566
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
SUM	5	199667	163698	209151	171474

TOTAL N	43	%INCREASE	.0475
TOTAL $	1892983	NEW TOTAL	1982900
AVERAGE	44023	NEW AVE.	46114
% TRS	.0898		
TRS $	169990	NEW TRS $	178064
T. W/TRS	2062973	T. W/TRS	2160964
AVE W/TRS	47976	AVE W/TRS	50255

BEST COPY AVAILABLE

	G	H	I	J	K
1	# TEACH	BA + 18	TOTAL	NEW BA+18	N. TOTAL
2					
3	0	30305	0	31744	0
4	0	32852	0	34412	0
5	0	35401	0	37083	0
6	0	36531	0	38266	0
7	0	37667	0	39456	0
8	0	38799	0	40642	0
9	3	39932	119796	41829	125486
10	0	41067	0	43018	0
11	1	42198	42198	44202	44202
12	5	43330	216650	45388	226941
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16					
17	9	378082	378644	396041	396630

BEST COPY AVAILABLE

# TEACH	MA/BA+36	total	n.MA/BA36	N. TOTAL
0	32286	0	33820	0
0	34836	0	36491	0
0	37385	0	39161	0
1	38517	38517	40347	40347
0	39649	0	41532	0
1	40784	40784	42721	42721
2	41916	83832	43907	87814
0	43048	0	45093	0
0	44179	0	46278	0
3	45313	135939	47465	142396
12	46447	557364	48653	583839
0	0	0	0	0
0	0	0	0	0
19	444360	856436	465467	897117

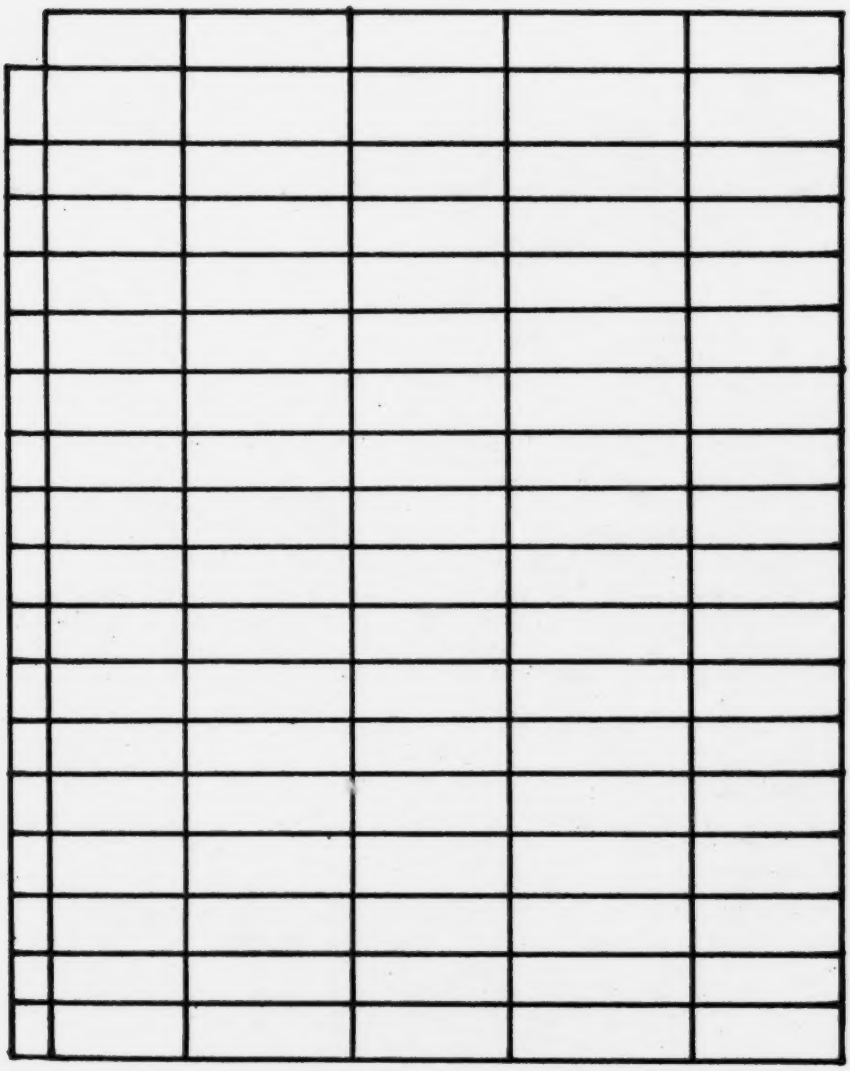
BEST COPY AVAILABLE

# TEACH	MA + 18	total	new MA+18
0	34269	0	35897
0	36817	0	38566
0	39367	0	41237
0	40499	0	42423
0	41633	0	43611
0	42765	0	44796
2	43896	87792	45981
0	45031	0	47170
0	46163	0	48356
0	47295	0	49542
1	48430	48430	50730
2	49562	99124	51916
0	0	0	0
5	515727	235346	540224

BEST COPY AVAILABLE

BEST COPY AVAILABLE

#	[illegible]	total	new Mat[illegible]	% total
0	36250	0	37972	0
0	38799	0	40642	0
0	41348	0	43312	0
0	42482	0	44500	0
0	43615	0	45687	0
0	44747	0	46872	0
0	45879	0	48058	0
0	47013	0	49246	0
1	48147	48147	50434	50434
0	49278	0	51619	0
0	50411	0	52806	0
0	51546	0	53994	0
4	52678	210712	55180	220721
5	592193	258859	620322	271155



115

```
>O4:/FI(M4*E19)+M4
>N4:/FL@SUM(L4*M4)
>M4:/FL34836
>L4:/FLO
>K4:/FI@SUM(G4*J4)
>J4:/FI(H4*E19)+H4
>I4:/FL@SUM(G4*H4)
>H4:/FL32852
>G4:/FLO
>F4:/FI@SUM(B4*E4)
>E4:/FI(C4*E19)+C4
>D4:/FL@SUM(B4*C4)
>C4:/FL30872
>B4:/FLO
>A4:/FL1
>Z3:/FI@SUM(V3*Y3)
>Y3:/FI(W3*E19)+W3
>X3:/FL@SUM(V3*W3)

 >X1:"  total
 >W1:"MA + 36
 >V1:" # TEACH
 >U1:" N. TOTAL
 >T1:"new MA+18
 >S1:"  total
 >R1:"MA + 18
 >Q1:" # TEACH
```

```
>W3:/FL36250
>V3:/FLO
>U3:/FI@SUM(Q3*T3)
>T3:/FI(R3*E19)+R3
>S3:/FL@SUM(Q3*R3)
>R3:/FL34269
>Q3:/FLO
>P3:/FI@SUM(L3*O3)
>O3:/FI(M3*E19)+M3
>N3:/FL@SUM(L3*M3)
>M3:/FL32286
>L3:/FLO
>K3:/FI@SUM(G3*J3)
>J3:/FI(H3*E19)+H3
>I3:/FL@SUM(G3*H3)
>H3:/FL30305
>G3:/FLO
>F3:/FI@SUM(B3*E3)
>E3:/FI(C3*E19)+C3
>D3:/FL@SUM(B3*C3)
>C3:/FL28322
>B3:/FL2
>A3:/FLO
>Z1:" N. TOTAL
>Y1:"new MA+36
```

```
>P1:" N. TOTAL
>O1:"n.MA/BA36
>N1:"  total
>M1:"MA/BA+36
>L1:"# TEACH
>K1:" N. TOTAL
>J1:"NEW BA+18
>I1:"TOTAL
>H1:"BA + 18
>G1:"# TEACH
>F1:"NEW TOTAL
>E1:"NEW BA+0
>D1:"  total
>C1:"  BA+0
>B1:"# TEACH
>A1:"YRS. EXP.
/W1
/GOC
/GRA
/GFI
/GC9
/X-/X>A1:>A1:
```


BEST COPY AVAILABLE

```
>R7:/FL41633
>Q7:/FL0
>P7:/FI@SUM(L7*O7)
>O7:/FI(M7*E19)+M7
>N7:/FL@SUM(L7*M7)
>M7:/FL39649
>L7:/FL0
>K7:/FI@SUM(G7*J7)
>J7:/FI(H7*E19)+H7
>I7:/FL@SUM(G7*H7)
>H7:/FL37667
>G7:/FL0
>F7:/FI@SUM(B7*E7)
>E7:/FI(C7*E19)+C7
>D7:/FL@SUM(B7*C7)
>C7:/FL35685
>B7:/FL1
>A7:/FL4
>Z6:/FI@SUM(V6*Y6)
>Y6:/FI(W6*E19)+W6
>X6:/FL@SUM(V6*W6)
>W6:/FL42482
>V6:/FL0
>U6:/FI@SUM(Q6*T6)
>T6:/FI(R6*E19)+R6
>S6:/FL@SUM(Q6*R6)
>R6:/FL40499
>Q6:/FL0
>P6:/FI@SUM(L6*O6)
>O6:/FI(M6*E19)+M6
>N6:/FL@SUM(L6*M6)
>M6:/FL38517
>L6:/FL1
>K6:/FI@SUM(G6*J6)
>J6:/FI(H6*E19)+H6
>I6:/FL@SUM(G6*H6)
>H6:/FL36531
>G6:/FL0
>F6:/FI@SUM(B6*E6)
>E6:/FI(C6*E19)+C6
>D6:/FL@SUM(B6*C6)
>C6:/FL34552
>B6:/FL1
>A6:/FL3
>Z5:/FI@SUM(V5*Y5)
>Y5:/FI(W5*E19)+W5
>X5:/FL@SUM(V5*W5)
>W5:/FL41348
>V5:/FL0
>U5:/FI@SUM(Q5*T5)
>T5:/FI(R5*E19)+R5
>S5:/FL@SUM(Q5*R5)
>R5:/FL39367
>Q5:/FL0
>P5:/FI@SUM(L5*O5)
>O5:/FI(M5*E19)+M5
>N5:/FL@SUM(L5*M5)
>M5:/FL37385
>L5:/FL0
>K5:/FI@SUM(G5*J5)
>J5:/FI(H5*E19)+H5
>I5:/FL@SUM(G5*H5)
>H5:/FL35401
>G5:/FL0
>F5:/FI@SUM(B5*E5)
>E5:/FI(C5*E19)+C5
>D5:/FL@SUM(B5*C5)
>C5:/FL33419
>R5:/FL0
>A5:/FL2
>Z4:/FI@SUM(V4*Y4)
>Y4:/FI(W4*E19)+W4
>X4:/FL@SUM(V4*W4)
>W4:/FL38799
>V4:/FL0
>U4:/FI@SUM(Q4*T4)
>T4:/FI(R4*E19)+R4
>S4:/FL@SUM(Q4*R4)
>R4:/FL36817
>Q4:/FL0
>P4:/FI@SUM(L4*O4)
```


BEST COPY AVAILABLE

```
>P10:/FI@SUM(L10*O10)
>O10:/FI(M10*E19)+M10
>N10:/FL@SUM(L10*M10)
>M10:/FL43048
>L10:/FL0
>K10:/FI@SUM(G10*J10
>J10:/FI(H10*E19)+H10
>I10:/FL@SUM(G10*H10
>H10:/FL41067
>G10:/FL0
>F10:/FI@SUM(B10*E
>E10:/FI(C10*E19)+C
>D10:/FL@SUM(B10*C10
>C10:/FL0
>B10:/FL0
>A10:/FL7
>Z9:/FI@SUM(V9*Y9)
>Y9:/FI(W9*E19)+W9
>X9:/FL@SUM(V9*W9)
>W9:/FL45879
>V9:/FL0
>U9:/FI@SUM(Q9*T9)
>T9:/FI(R9*E19)+R9
>S9:/FL@SUM(Q9*R9)
>R9:/FL43896
>Q9:/FL2
>P9:/FI@SUM(L9*O9)
>O9:/FI(M9*E19)+M9
>Y8:/FI(W8*E19)+W8
>X8:/FL@SUM(V8*W8)
>W8:/FL44747
>V8:/FL0
>U8:/FI@SUM(Q8*T8)
>T8:/FI(R8*E19)+R8
>S8:/FL@SUM(Q8*R8)
>R8:/FL42765
>Q8:/FL0
>P8:/FI@SUM(L8*O8)
>O8:/FI(M8*E19)+M8
>N8:/FL@SUM(L8*M8)
>M8:/FL40784
>L8:/FL1
>K8:/FI@SUM(G8*J8)
>J8:/FI(H8*E19)+H8
>I8:/FL@SUM(G8*H8)
>H8:/FL38799
>G8:/FL0
>F8:/FI@SUM(B8*E8)
>E8:/FI(C8*E19)+C8
>D8:/FL@SUM(B8*C8)
>C8:/FL36817
>B8:/FL1
>A8:/FL5
>Z7:/FI@SUM(V7*Y7)
>Y7:/FI(W7*E19)+W7
>N9:/FL@SUM(L9*M9)
>M9:/FL41916
>L9:/FL2
>K9:/FI@SUM(G9*J9)
>J9:/FI(H9*E19)+H9
>I9:/FL@SUM(G9*H9)
>H9:/FL39932
>G9:/FL3
>F9:/FI@SUM(B9*E9)
>E9:/FI(C9*E19)+C9
>D9:/FL@SUM(B9*C9)
>C9:/FL0
>B9:/FL0
>A9:/FL6
>Z8:/FI@SUM(V8*Y8)
>X7:/FL@SUM(V7*W7)
>W7:/FL43615
>V7:/FL0
>U7:/FI@SUM(Q7*T7)
>T7:/FI(R7*E19)+R7
>S7:/FL@SUM(Q7*R7)
```


BEST COPY AVAILABLE

```
>R13:/FL48430
>Q13:/FL1
>P13:/FI@SUM(L13*Q1
>O13:/FI(M13*E19)+M
>N13:/FL@SUM(L13*M1
>M13:/FL46447
>L13:/FL12
>K13:/FI@SUM(G13*J
>J13:/FI(H13*E19)+H
>I13:/FL@SUM(G13*H1
>H13:/FL0
>G13:/FL0
>F13:/FI@SUM(B13*E1
>E13:/FI(C13*E19)+C
>D13:/FL@SUM(B13 _1
>C13:/FL0
>B13:/FL0
>A13:/FL10
>Z12:/FI@SUM(V12*Y
>Y12:/FI(W12*E19)+W
>X12:/FL@SUM(V12*W
>W12:/FL49278
>V12:/FL0
>U12:/FI@SUM(Q12*T12)
>T12:/FI(R12*E19)+R12
>S12:/FL@SUM(Q12*R12)
>R12:/FL47295
>Q12:/FL0
>P12:/FI@SUM(L12*Q1
>O12:/FI(M12*E19)+M
>N12:/FL@SUM(L12*M
>M12:/FL45313
>L12:/FL3
>K12:/FI@SUM(G12*J12
>J12:/FI(H12*E19)+H12
>I12:/FL@SUM(G12*H12
>H12:/FL43330
>G12:/FL5
>F12:/FI@SUM(B12*E
>E12:/FI(C12*E19)+C
>D12:/FL@SUM(B12*C1
>C12:/FL0
>B12:/FL0
>A12:/FL9
>Z11:/FI@SUM(V11*Y
>Y11:/FI(W11*E19)+W
>X11:/FL@SUM(V11*W
>W11:/FL48147
>V11:/FL1
>U11:/FI@SUM(Q11*T
>T11:/FI(R11*E19)+R
>S11:/FL@SUM(Q11*R
>R11:/FL46163
>Q11:/FL0
>P11:/FI@SUM(L11*Q1
>O11:/FI(M11*E19)+M
>N1 [illegible]
>M11:/FL44179
>L11:/FL0
>K11:/FI@SUM(G11*J1
>J11:/FI(H11*E19)+H
>I11:/FL@SUM(G11*H1
>H11:/FL42198
>G11:/FL1
>F11:/FI@SUM(B11*E1
>E11:/FI(C11*E19)+C
>D11:/FL@SUM(B11*C1
>C11:/FL0
>B11:/FL0
>A11:/FL8
>Z10:/FI@SUM(V10*Y1
>Y10:/FI(W10*E19)+W
>X10:/FL@SUM(V10*W1
>W10:/FL47013
>V10:/FL0
>U10:/FI@SUM(Q10*T10)
>T10:/FI(R10*E19)+R10
>S10:/FL@SUM(Q10*R10)
>R10:/FL45031
>Q10:/FL0
```


BEST COPY AVAILABLE

```
>U16:"---------      >T15:/FI(R15*E19)+R1      >S14:/FL@SUM(Q14*R14)
>T16:"---------      >S15:/FL@SUM(Q15*R1       >R14:/FL49562
 S16:"----------     >R15:/FLO                  Q14:/FL2
>R16:"----------     >Q15:/FLO                 >P14:/FI@SUM(L14*O14
>Q16:"----------     >P15:/FI@SUM(L15*O1       >O14:/FI(M14*E19)+M1
 P16:"----------     >O15:/FI(M15*E19)+M1      >N14:/FL@SUM(L14*M14)
>O16:"----------     >N15:/FL@SUM(L15*M1       >M14:/FLO
>N16:"----------     >M15:/FLO                 >L14:/FLO
>M16:"----------     >L15:/FLO                 >K14:/FI@SUM(G14*J14)
>L16:"----------     >K15:/FI@SUM(G15*J15)     >J14:/FI(H14*E19)+H14
>K16:"----------     >J15:/FI(H15*E19)+H1      >I14:/FL@SUM(G14*H14)
>J16:"----------     >I15:/FL@SUM(G15*H1       >H14:/FLO
>I16:"----------          >H15:/FLO            >G14:/FLO
>H16:"----------          >G15:/FLO            >F14:/FI@SUM(B14*E14)
>G16:"----------          >F15:/FI@SUM(B15*E1   >E14:/FI(C14*E19)+C14
>F16:"----------          >E15:/FI(C15*E19)+C1  >D14:/FL@SUM(B14*C14)
>E16:"----------          >D15:/FL@SUM(B15*C1   >C14:/FLO
>D16:"----------          >C15:/FLO            >B14:/FLO
>C16:"----------          >B15:/FLO            >A14:/FL11
>B16:"----------          >A15:/FL12           >Z13:/FI@SUM(V13*Y13)
>A16:"----------          >Z14:/FI@SUM(V14*Y14) >Y13:/FI(W13*E19)+W13
>Z15:/FI@SUM(V15*Y1       >Y14:/FI(W14*E19)+W14 >X13:/FL@SUM(V13*W13)
>Y15:/FI(W15*E19)+W1      >X14:/FL@SUM(V14*W14) >W13:/FL50411
>X15:/FL@SUM(V15*W15)     >W14:/FL51546         >V13:/FLO
>W15:/FL52678             >V14:/FLO             >U13:/FI@SUM(Q13*T13)
>V15:/FL4                 >U14:/FI@SUM(Q14*T14) >T13:/FI(R13*E19)+R13
>U15:/FI@SUM(Q15*T15)     >T14:/FI(R14*E19)+R1  >S13:/FL@SUM(Q13*R13)
```


BEST COPY AVAILABLE

```
>E26:(E25/B19)
>D26:"AVE W/TRS
>B26:(B25/B19)
>A26:"AVE W/TRS
>E25:(B23*E20)+E20
>D25:"T. W/TRS
>B25:(B23*B20)+B20
>A25:"T. W/TRS
>E24:(B23*E20)
>D24:"NEW TRS $
>B24:(B20*B23)
>A24:"TRS $
>B23:/FG.0898
>A23:"% TRS
>E21:(E20/B19)
>D21:"NEW AVE.
>B21:@SUM(B20/B19)
>A21:"AVERAGE
>E20:/FI@SUM(F17+K17+P17+U17+Z17)
>D20:"NEW TOTAL
>B20:/FL@SUM(D17+I17+N17+S17+X17)
>A20:"TOTAL $
>J19:/FG
>E19:/FG.0475
>D19:"%INCREASE
>B19:/FL@SUM(B17+G17+L17+Q17+V17)
>A19:"TOTAL N
>Z17:/FI@SUM(Z3...Z15)
>Y17:/FI@SUM(Y3...Y15)
>X17:/FL@SUM(X3...X15)
>W17:/FL@SUM(W3...W15)
>V17:/FL@SUM(V3...V15)
>U17:/FI@SUM(U3...U15)
>T17:/FI@SUM(T3...T15)
>S17:/FL@SUM(S3...S15)
>R17:/FL@SUM(R3...R15)
>Q17:/FL@SUM(Q3...Q15)
>P17:/FI@SUM(P3...P15)
>O17:/FI@SUM(O3...O15)
>N17:/FL@SUM(N3...N15)
>M17:/FL@SUM(M3...M15)
>L17:/FL@SUM(L3...L15)
>K17:/FI@SUM(K3...K15)
>J17:/FI@SUM(J3...J15)
>I17:/FL@SUM(I3...I15)
>H17:/FL@SUM(H3...H15)
>G17:/FL@SUM(G3...G15)
>F17:/FI@SUM(F3...F15)
>E17:/FI@SUM(E3...E15)
>D17:/FL@SUM(D3...D15)
>C17:/FL@SUM(C3...C15)
>B17:/FL@SUM(B3...B15)
>A17:"SUM
>Z16:"---------
>Y16:"---------
>X16:"---------
>W16:"---------
>V16:"---------
```


BEST COPY AVAILABLE

STATEWIDE RANKING OF FY-85 TEACHER BASE SALARIES, INCLUDING OTHER BENEFITS

BASE SALARY	BASE PER DAY	DISTRICT	REGION	TYPE	# DAYS	% HEALTH PAID	% PAID DEPEND	% PAID DENTAL VISUAL	% PAID DEPEND D/V	PERSONAL LEAVE DAYS PER/TEA
27413	145.813	ADAK	SW	REAA	188	100	0	100	0	6
27360	145.531	ALASKA GATEWAY	INT	REAA	188	100	100	100	100	4
31329	163.171	ALEUTIAN REGION	SW	REAA	192	100	100	100	100	8
25075	134.090	ANCHORAGE	SC	BOR	187	100	100	100	100	7
25795	137.207	ANNETTE ISLAND	SE	REAA	188	100	100	100	100	3
29500	153.645	BERING STRAIT	NW	REAA	192	100	100	100	100	4
28633	152.303	BRISTOL BAY	SW	BOR	188	100	0	100	0	2
24100	128.191	CHATHAM	SE	REAA	188	100	100	100	100	10
24000	126.984	CHUGACH	SC	REAA	189	100	100	100	100	10
0	0.000	COPPER RIVER	INT	REAA	189	100	100	100	100	2
25481	135.537	CORDOVA	SE	C	188	100	100	100	100	4
24700	133.513	CRAIG	SE	C	185	100	100	100	100	3
28620	152.234	DELTA-GREELY	INT	REAA	188	100	100	100	100	4
29140	154.179	DILLINGHAM	SW	C	189	100	100	100	100	6
26489	140.898	FAIRBANKS	INT	BOR	188	100	100	100	100	5
32664	173.744	GALENA	INT	C	188	100	100	100	100	5
26624	142.374	HAINES	SE	BOR	187	100	100	100	100	5
24239	127.573	HOONAH	SE	C	190	100	100	100	100	1
25000	133.689	HYDABURG	SE	C	187	100	100	100	100	3
27476	146.148	IDITAROD	INT	REAA	188	100	100	100	100	6
27036	146.140	JUNEAU	SE	BOR	185	100	100	100	100	7
24835	131.402	KAKE	SE	C	189	100	100	100	100	7
25504	135.659	KENAI PENINSULA	SC	BOR	188	100	100	100	100	7
25988	138.973	KETCHIKAN	SE	BOR	187	90	90	90	90	5
29300	154.210	KING COVE	SW	C	190	100	100	100	100	8
24300	127.894	KLAWOCK	SE	C	190	100	100	100	100	2
24991	132.930	KODIAK ISLAND	SW	BOR	188	100	100	100	100	5
26738	142.223	KUSPUK	SW	REAA	188	100	100	100	100	6

```
. display structure
STRUCTURE FOR FILE:   C:SALARY5 .DBF
NUMBER OF RECORDS:    00053
DATE OF LAST UPDATE:  02/18/85
PRIMARY USE DATABASE
FLD        NAME        TYPE  WIDTH    DEC
001      DIST           C     017
002      TYPE           C     004
003      REGION         C     003
004      ADM:85         N     005
005      BASE2          N     005
006      BASE4          N     005
007      BASE5          N     005
008      BASE6          N     005
009      PERIBASE5      N     005     002
010      PERIBASE6      N     005     002
011      PERIB24        N     005     002
012      CPI24          N     005     002
013      TOP2           N     005
014      TOP4           N     005
015      TOP5           N     005
016      TOP6           N     005
017      PERITOP5       N     005     002
018      PERITOP6       N     005     002
019      PERITOP24      N     005     002
020      MODEDOL        N     005
021      DAYPAYMODE     N     007     003
022      AVESAL         N     005
023      MODECOL        C     014
024      MODESTEP       N     002
025      NO:TEAFTE      N     009     002
026      NO:DAYS        N     003
027      DAYPAYB        N     007     003
028      DAYPAYT        N     007     003
** TOTAL **                 00164
```

```
*                          UPDATE.CMD

* This file is the master file that is used to produce the monthly report.
* The only thing that the operator needs to do is to update the records in
* the following  data bases:

* DATA BASE FILE !  DESCRIPTION/FUNCTION
*    - Salary5       ! Contains all the historical increases such as, salary
*                    ! figues FY-84-82 and benefit amounts.

*    - Salary5a      ! Contains all in salary schedule information for FY-85.
*                    !

*    - SALARY5b      ! Contains all the salary information in relation to number
*      ! of teachers at which salary ranges and estimated
*              ! total salary costs.
*
SET MARGIN TO 0

USE SALARY5

* These commands automatically compute the percentage of increases from year
* to year for the top and base salaries.

REPLACE ALL DAYPAYT WITH (TOP5/NO:DAYS) FOR TOP5)10

REPLACE ALL DAYPAYB WITH (BASE5/NO:DAYS) FOR TOP5)10

REPLACE ALL PERITOP5 WITH ((TOP5-TOP4)/TOP4)*100 FOR TOP5)10

REPLACE ALL PERITOP6 WITH ((TOP6-TOP5)/TOP5)*100 FOR TOP6)10

REPLACE ALL PERIBASE5 WITH ((BASE5-BASE4)/BASE4)*100 FOR BASE5)10

REPLACE ALL PERIBASE6 WITH ((BASE6-BASE5)/BASE5)*100 FOR BASE6)10

REPLACE ALL PERIB24 WITH ((BASE4-BASE2)/BASE2)*100 FOR BASE4)10

REPLACE ALL PERITOP24 WITH ((TOP4-TOP2)/TOP2)*100 FOR TOP4)10
```

```
* These commands automatically compute the index, daily rates of pay and the
* schedule average for each district and puts the information in each record.

REPLACE ALL TL:TEA:LV WITH (PERSONAL+OTHERLV)

REPLACE ALL INDEX WITH (TOP5/BASE5) FOR TOP5)10


REPLACE ALL SCHD:AVE WITH (BASE5+TOP5)/2 FOR TOP5)10

REPLACE ALL DAYPAYAV WITH (AVESAL/NO:DAYS) FOR AVESAL )0
```


BEST COPY AVAILABLE

```
* This command create an index for FY-85 top salaries and stores it in
* an index file called TOPSAL.NDX.

USE  SALARY5A
INDEX ON STR(TOP5,5) + DIST TO TOPSAL

* This command uses the topsal index and and instructs the computer to
* print it out in the report form called topsal.frm.  The report form
* tells the computer how to print out the form.

USE  SALARY5A INDEX TOPSAL
REPORT FORM TOPSAL TO PRINT

* This command create an index for FY-85 top salaries and stores it in
* an index file called REGIONT.NDX.  This file is indexed by region +
* top salary.

USE  SALARY5A
INDEX ON REGION +STR(TOP5,5) TO REGIONT

* This command uses the REGIONT index and instructs the computer to
* print it out in the report form called, REGIONT.FRM.  The report form
* tells the computer how to print out the form.


USE  SALARY5A INDEX REGIONT
REPORT FORM REGIONT TO PRINT
```

COPY AVAILABLE

```
*This command releases all memory variables.

RELEASE ALL

* This command tells the computer to go do the instructions in the file
* called TSALARY.CMD.

DO TSALARY

* This command tells the computer to go do the instructions in the file
* called TSALINCR.CMD.

DO TSALINCR



USE SALARY5A

* This command updates the  SALARY5A fields from the  SALARY5 data base file

UPDATE ON DIST FROM  SALARY5 REPLACE BASE5
UPDATE ON DIST FROM  SALARY5 REPLACE TOP5
UPDATE ON DIST FROM  SALARY5 REPLACE DAYPAYB
UPDATE ON DIST FROM  SALARY5 REPLACE AVESAL
UPDATE ON DIST FROM  SALARY5 REPLACE ADM:85
UPDATE ON DIST FROM  SALARY5 REPLACE DAYPAYT
UPDATE ON DIST FROM  SALARY5 REPLACE NO:DAYS
UPDATE ON DIST FROM  SALARY5 REPLACE NO:TEAFTE
```

PLATO and EDNET

A GROUP PRESENTATION FOR THE AACED CONVENTION, APRIL 4-6, 1985

Ray Lehrman, Moderator

INTRODUCTION

The Alaska Rural Education Network (EDNET) Project was developed by Alaska Pacific University and funded by the Legislature to install and support the use of PLATO Computer-Based Education hardware and software in four rural Alaskan schools for the 1984-1985 school year.

Three speakers will acquaint you with the essential details. Their presentations will be moderated by Mr. Will Files, an EDNET Project PLATO-using teacher from Ft. Yukon School.

THE TOPICS AND THE SPEAKERS

I. WHAT IS THE PLATO COMPUTER-BASED EDUCATIONAL SYSTEM? (12 minutes)

by: Dr. Jan Ingram, Product Manager for Education Services, Control Data Corporation, Anchorage office

II. HOW DOES THE ALASKA EDNET PROJECT USE PLATO? (12 minutes)

by: Dr. Ray Lehrman, EDNET Project Director, Professor of Education, Alaska Pacific University

III. WHAT IS OUZINKIE SCHOOL DOING WITH PLATO? (12 minutes)

by: Ms. Patt Gibbs, EDNET PLATO-using teacher from Ouzinkie School, Kodiak Island Borough School District

IV. QUESTIONS AND ANSWERS (remainder of session)

by: Will Files, Jan Ingram, Ray Lehrman, Patt Gibbs, and the audience.

POSTSECONDARY TRAINING OPTIONS

THROUGH THE ALASKA DEPARTMENT OF EDUCATION

PRESENTER: Robert J. McCabe, Ed.D.
Office Occupations Department Head
Alaska Vocational Technical Center (AVTEC)
Box 615
Seward, Alaska 99664 (907) 224-3322, Ext. 233

INTRODUCTION: This presentation was requested by the AACED officers to provide the conference participants with information about postsecondary training options at AVTEC in the area of computer applications. The presentation should be useful to individuals who are planning to attend computer-related courses this summer. The presentation should also help school personnel who would like information to relay to others. Information will relate to the summer program for school personnel with limited reference to the fall program. The fall program is intended for students preparing for office careers.

PRESENTATION CONTENT: The presentation will provide information on hardware/software available at AVTEC, personal software options and instructional methods for the summer program, graduate credit available through the University of Alaska (VE 693 ST), housing, transportation, and recreational opportunities in Seward and the Kenai Peninsula area, etc. AVTEC has 20 IBM PC-XTs, 3 IBM Displaywriters, 3 Wangs, and 5 Apple II+ and IIe computers available. The hardware/software selection process and training experiences will also be discussed if participants are interested. Refreshments and a short video tape on Seward and AVTEC will be available.

This presentation is specifically related to the Alaska Vocational Technical Center in Seward. AVTEC is a postsecondary technical school operated by the Alaska Department of Education.

Alan McCurry and Niki McCurry

IF THE SHOE FITS THEY'LL WEAR IT: PROJECT APEL, A HOLISTIC PROCESS ORIENTED APPROACH ENABLING EVERY TEACHER AT EVERY SCHOOL TO INTEGRATE THE USE OF COMPUTERS INTO THE REGULAR SCHOOL CURRICULUM K-12.

As the microcomputer becomes less of a novelty and more of a tool, more and more teachers and administrators in Alaska are now turning from a strict concern with computer literacy and programming to asking questions such as: " How can we integrate this useful educational tool into our traditional academic curriculum?" The proposed presentation will review the process by which the Yukon Koyukuk School District has attempted the complete integration of computers into a traditional curriculum area so that every teacher in every classroom now uses computers on a regular basis to teach almost every component of that subject area. The detailed process is described much more fully in the enclosed APEL PROJECT HANDBOOK.

PROJECT APEL is a microcomputer-focused bilingual language arts project in the Yukon-Koyukuk School District. It is now in its third year and showing impressive gains in teacher and student behaviors. It may offer a model for demonstrating the process of integrating computers into a basic curriculum area on a district-wide basis.

Unlike many computer projects which focus on one possible use of the computer for teaching one skill, such as Logo for creativity, word processing for fluency, drills for grammar practice, PROJECT APEL uses a holistic approach to computer use. Computer use, through appropriately selected courseware/software, is integrated into every aspect of the YKSD LANGUAGE ARTS/READING CURRICULUM. All components of the program and most techniques upon which teachers have traditionally relied for teaching those components have been computer-enhanced.

For example, the penmanship program has been up-dated to include keyboarding skills with recommended software such as Typing Tutor. Traditional "seat work" grammar drills have been improved through the use of systematic and interactive grammar programs to teach each instructional objective in the curriculum guide. The writing process and involvement in the Alaska State Writing Project has been facilitated through the use of word processors by 100% of YKSD students. Reading for comprehension is supplemented through a variety of "reading lab" type programs; reading for pleasure is encouraged through interactive fiction and software programs which supplement good children's literature.

This holistic integration has been achieved through a multi-phase, multi-year process which includes 1) A computer-based curriculum needs assessment; 2) A procedure for materials selection and acquisition; 3) Materials development, and 4) staff training. It is this process which the proposed presentation will address.

In order to facilitate the curriculum integration and training components of PROJECT APEL and to publish practical information about the APEL model which other school districts might use in integrating computers into their own district curricula, APEL has published the APEL HANDBOOK. This handbook may serve not only as a model for a computer integrated language arts curriculum but as a model for computer integration into other curriculum areas as well. Indeed, the APEL HANDBOOK has been adopted by the Fairbanks North Star Borough as a model with the intention, on the part of Computer Coordinator Karen Parr (who was one of the co-authors), to similarly integrate another subject area.

Niki McCurry and Alan McCurry

ALASKAN SOFTWARE FOR ALASKAN STUDENTS: DEVELOPING COURSEWARE TO MEET THE NEEDS OF NATIVE ALASKANS

PROJECT APEL, a bilingual microcomputer project at the Yukon-Koyukuk School District (funded by TITLE VII), has been engaged, for the last three years in one of the most exciting and ground breaking efforts in the field of bilingual education: the development of a computer courseware series to meet the specific English language learning needs of bilingual Native American students.

Each computer courseware kit in the INTERIOR ATHABASKAN ENGLISH GRAMMAR SERIES addresses one specific instructional objective. The list of instructional objectives addressed in the SERIES is based upon a five year longitudinal research study (begun in 1982) of the impact of an Athabaskan language, Koyukon, on the English patterns in the Yukon Koyukuk Region. From samples of student written and oral language, a list of high frequency "errors" or differences from Standard American English is developed. This list is then analyzed in the light of information about the original Native language in the region, Koyukon Athabaskan (CF. KOYUKON SCOPE & SEQUENCE, YKSD, 1983). Unique features of Koyukon-English are identified. In addition, "errors" which are not unique, but of persistent high frequency in student samples are also noted. These features are then targeted as instructional objectives for the series of supplementary computer lessons. A copy of the first two research studies is available as a bound volume, APEL RESEARCH REPORTS VOLUME I (YKSD 1984).

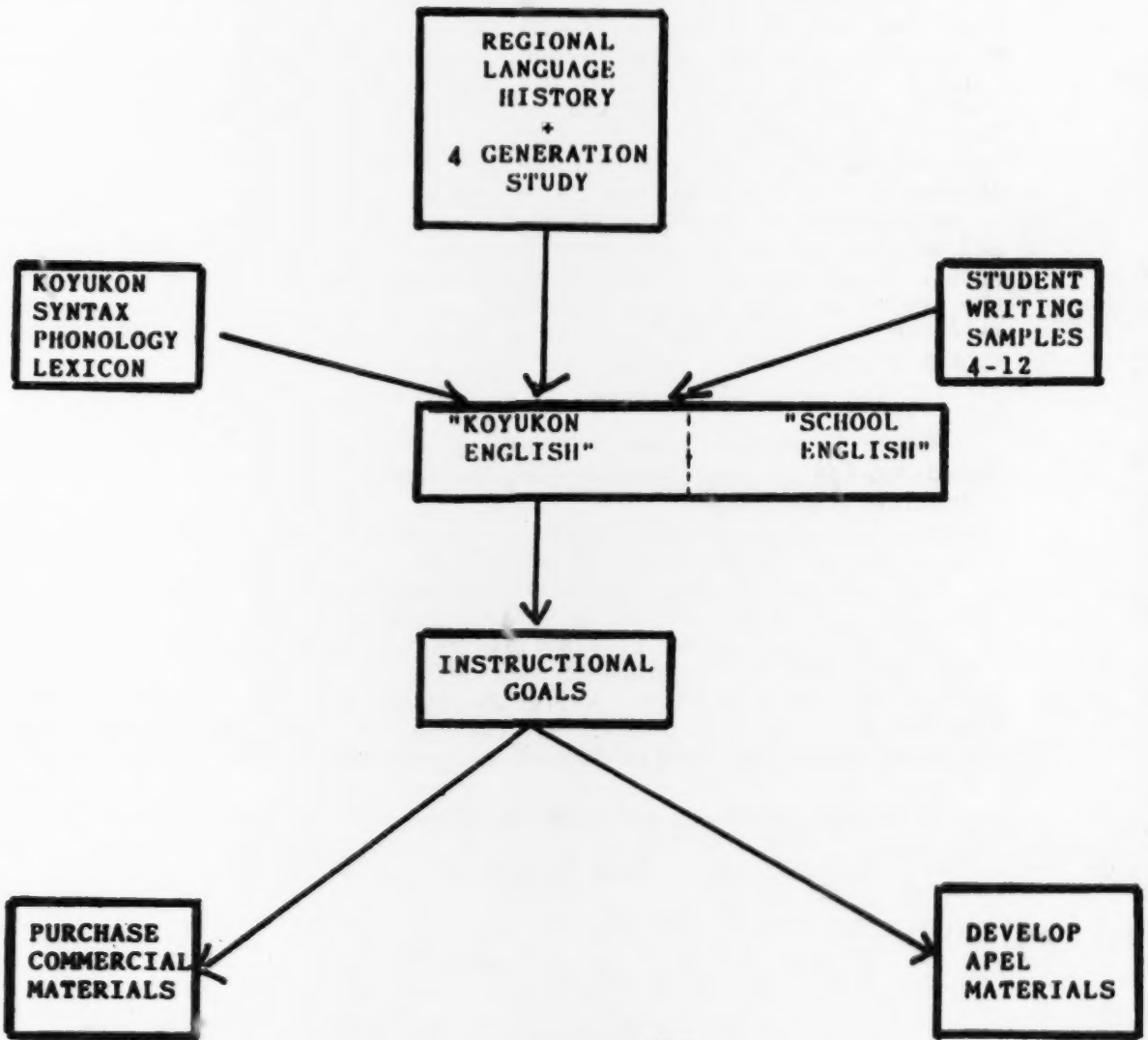
The kits are developed by experienced Alaskan rural educators under the direction of the PROJECT APEL Directors, who designed the series. Linguistic consultation was provided by the University of Alaska, Alaska Native Language Center and Department of Cross-Cultural Communication. Each kit addresses the specified instructional objective in the context of an interactive "adventure" which takes place in interior rural Alaska. The series as a whole is meant to supplement, not replace, the basic English proficiency curriculum. The computer kits are meant to supplement, not replace, teacher instruction and student writing activites. While the kits are based on research into a specific dialect, Koyukon-English, they may be appropriate for bilingual children from other related Athabaskan languages.

Six kits have been developed to date:

1. SKY PILOT: articles
2. TREASURE HUNT: verb tenses
3. RIVER RUNNER: prepositions
4. IDITAROD MUSHER: verb forms
5. MYSTERY MISSION: adjective/adverb
6. DENEKI & DINEEGA: homophone "gremlins"

The PROJECT APEL INTERIOR ATHABASKAN ENGLISH GRAMMAR COURSEWARE SERIES is innovative both because it utilizes the capabilities of the microcomputer in a hitherto un-developed area, bilingual education, and because the linguistic research upon which the software is based is unprecedented in Alaska and the lower-forty-eight.

APEL INSTRUCTIONAL IMPROVEMENT PROCESS



MATCHING COMPUTER SOFTWARE TO THE ELEMENTARY SCHOOL CURRICULUM

by Louise Petermann, Coordinator, Math/Microcomputer Education
Anchorage School District

Computer education in the Anchorage School District emphasizes using the computer as a problem solving tool capable of being used by everyone. The Computer Instructional Program is not aimed at training people for computer careers. The District has secondary vocational education and Career Center courses designed for that purpose. Rather than looking at the computer as an object of and in itself, the Computer Instructional Program takes a broader view of how the use of a computer can help the individual solve problems in their daily lives. The District's program introduces students to applications including problem solving activities, word processing activities, data processing activities, and educational activities.

An educational plan for the use of computer technology in the Anchorage School District, grades K-12, was developed in December, 1979, by a district-wide committee. Six years later, a computer insructional program has been implemented, over 1700 classroom teachers have participated in credit courses on educational applications of microcomputers, a K-12 Computer Education Scope and Sequence has been developed, and the School Board has allocated funds to insure that there is at least one microcomputer for every elementary teaching station and that there is at least one computer lab in every secondary school in the District.

The development of a K-12 Computer Education Scope and Sequence has contributed to the integration of new technology throughout the curriculum. A thirty member Ad Hoc Microcomputer Curriculum Committee composed of elementary and secondary classroom teachers, principals, and members of the community reviewed goals and objectives from other school districts and states, read the latest research, and defined what they felt Anchorage students needed to know about computers and computer applications. The committee developed a program philosophy, K-12 goals and objectives which can be infused into all curriculum areas, and identified software and other resources to achieve K-6 grade level objectives for language arts, mathematics, science, and social studies. The focus of the scope and sequence is to view the computer as a teaching/learning/problem solving tool rather than as the object of instruction and to demonstrate how to integrate computer skills throughout the curriculum.

Grade level documents have been developed and distributed for each grade, K-6. These documents contain not only objectives and suggested software for each grade level, but also a series of Software Teaching Suggestions. The Software Teaching Suggestions contain specific computer objectives, specific curriculum objectives, and ideas on how, when, and where a specific piece of software could be used. The elementary portion of the Computer Education Scope and Sequence is currently being validated by classroom teachers. Revisions will be made in the document based upon recommendations from these teachers.

The District's elementary math scope and sequence documents also assist teachers with integrating computers into the curriculum. Each grade level document identifies computer software which matches the math objectives of that grade level.

The process which has been used to identify software which reinforces the District's curriculum goals has been one of first identifying good software and then matching it to the curriculum. This process is currently starting to reverse itself. That is, teachers are now asking for software to meet specific goals and objectives. To facilitate the process of finding software to match curriculum objectives, data bases are being developed and printouts are being provided which identifies existing software by subject area, topic, and grade level.

One-credit courses, inservices, and workshops designed to facilitate the integration of computers into the curriculum are being provided. Contained within these staff development activities is the opportunity for teachers to develop their own Software Teaching Suggestions. These are being collected and will be compiled, edited, and distributed in the fall of 1985.

THE USE OF COMMERCIALLY AVAILABLE UTILITY PROGRAMS BY THE STUDENT PROGRAMMER

Don Putnam

There are two schools of thought on the teaching of programming. The first school emphasizes a total student effort from first to last, with the student constructing from scratch all subroutines and utilities.

The second school holds that while original programming and an individual concept are important requirements for the student programmer, it is not necessary to continually reinvent the wheel. A good example would be an original program requiring a sort routine. Sort routines that run in Basic are notoriously slow, and conceptually difficult for the beginning programmer. A good, fast sorting routine would enable the student to truly demonstrate the worth of his/her program.

Utility programs have existed on the software market from the start. Hackers have delighted in adopting and modifying utilities into their own products. The practice of typing in a listing from a magazine was a short-lived phenomenon (although there are still vestigal listings available in some magazines; more about this later). Programmers were willing to purchase utilities for reasons of time-saving and bug-free programs; listings were notorious for typographical errors, omitted lines in the article, let alone often someone attempted to type in 2000 lines of someone else's thinking.

There are several areas of computer programming that are very amenable to the use of commercially available utilities; here are some of the more important:

1. Line Editing - The differences in the ease of line editing vary from one computer to the next; if it is a problem, a good line editor program can be of immense help to the novitiate.

2. Graphics and Shape Tables - Many student programmers want to include graphics as part of their program. Before the advent of usuable graphics programs, programmers were discouraged from attempting a graphics to avoid them from becoming bogged down. There are excellent programs now that allow them to create their own graphics and place them in their program.
3. Type Fonts - Type fonts, existing both as screen displays and screen dumps, are invaluable to the beginning programmer. An extremely complex application of high resolution graphics, type fonts provide the extra flexibility to turn a bland ASCII text title page into an attractive eye-catching display.
4. Data Management Utilities - This area has already been touched upon; for beginning programmers, it is an obvious advantage to free up the programmer's thinking.
5. Menus - The use of menus found in utility programs is a borderline issue with me. A simple menu program is not hard to construct (I think one of my teacher programming classes might disagree) and is a useful practice to further refine programming skills. For myself, I think the deciding factor would be the number of programs to be called, and the complexity of formatting. Menu programs would allow the student programmer to concentrate on his other programs. Perhaps the programmer should be made to demonstrate mastery of menu programming before being allowed to use a commercial product.

The above types of utilities should be used judiciously to allow the programmer to develop in a sound fashion. To the purists who would allege that they are a needless crutch, I just say, "pooh"! A better analogy would

training wheels. In an interesting trivia study, it turned up that over 75% of the American cycling medalists learned to ride a bicycle using training wheels. What more is needed to add?

BASIC LITERACY THROUGH MICROCOMPUTERS (BLTM)

TEACHING COMPUTER KEYBOARDING, READING, SPELLING AND TYPING SKILLS

Ethna R. Reid, Reid Foundation
3310 South 2700 East
Salt Lake City, Utah 84109

BLTM is a program that will enhance reading and spelling achievement because typing skills are introduced to students at the same time as the language skills. Without an ability to use a computer keyboard correctly, the reading act would be unnecessarily delayed. Using the "touch" system to respond on a computer allows students to become fluent in reading as well as in typing. With effective keyboard skills, the computer becomes a second teacher for students.

The authors, Marvis Collett, Charles Parker and Ethna Reid, focused their expertise in reading and typing instruction and computer programming on the development of a highly innovative and motivating instructional method.

BLTM teaches:

-- Computer keyboarding, typing, reading, spelling and visual sequential memory skills.
-- Use of special computer keys.
-- Word recognition and comprehension skills.
-- Typing skills so that fluent reading is not deterred by an inability to use the computer. (A student is more motivated to learn typing skills on a computer than on a typewriter. It is also more controlled--since an incorrectly hit key will not depress.)
-- "Home row" keys of the computer first, so the introduction of reading skills follows the sequence of those letters first.
-- Letter names and sounds for letters and combinations of letters.

-- Oral reading rate without sacrificing reading accuracy.

-- Varying reading rates.

-- Spelling words that are learned in reading.

-- Auditory discrimination and memory skills.

-- Silent reading comprehension and rate.

-- Mechanics of writing: capitalization, quotation marks, semicolons, periods, commas.

-- Ability to concentrate and sustain attention on a task. Retention is enhanced with the multiple practices.

BLTM evaluates:

-- Computer skills (A test that can be used as a pretest and post test of the computer is enclosed in the program package.)

-- Typing skills (A test that can be used as a pretest and post test of the typing skills is enclosed in the program package.)

-- Reading and spelling skills through:

Mastery tests that occur after every nine lessons--Lessons 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. These can be administered as pretests for each unit of nine lessons. The mastery tests measure typing accuracy through a dictation test of letter names, sounds, words, phrases and sentences; typing accuracy and rate as students type stories from the computer screen; reading and spelling accuracy.

A comparison of a student's oral and silent reading rate.

-- Visual memory match and recall skills as well as reading and spelling rate.

BLTM reinforces a student's:

-- Interest in learning--using the computer is motivating.

-- Self-esteem. Skills are introduced sequentially and practice is provided

as needed so mastery is obtained readily. Successful completion of each lesson is rewarded with data of accuracy printed at the bottom of the screen, an animated picture, the word "WOW" and a tune. Students are also rewarded for passing mastery tests in a similar manner. During a lesson, if an incorrect letter is typed, the computer key will not depress and a bell sounds to alert the learner. This is not true during a mastery test. At the conclusion of a mastery test, words to be studied appear on the screen.

-- Previously obtained reading, spelling, typing and computer skills.

The program is packaged in a looseleaf binder. Included in the binder are:

1. Four computer disks
   a. Lessons and Mastery Tests
   b. Paced Reading Lessons (These occur with the lesson that precedes each mastery test. The lessons are presented at four different speeds to be selected by the student.)
   c. Timed Practices (These also occur with each lesson prior to the mastery test.)
      (1) Visual Memory Match exercises in which the student is required to wait and remember the word that is flashed on the computer screen and then type it.
      (2) Visual Timed Recall exercises that are presented at 33 different timed presentations. The time presentation of the word automatically increases or decreases in rate depending on the student's prior accuracy.
2. Instructions for computer use; for teaching typing, names of letters, sounds, words and the reading of stories; for teaching spelling and

visual memory skills; for testing comprehension.

3. Printed stories. (These are read by the student with each lesson prior to a mastery test.)
4. Typing drills to be used with the computer on or off. With the computer on, and using a mastery test lesson, the student can see his/her typed copy on the computer screen. With the computer off, the student can practice his/her fingering, watching the printed copy.
5. A scope and sequence chart of skills and vocabulary words introduced in the 100 lessons.
6. A sample Pupil Record Form to be reproduced for each student. The student records the dates he passes each lesson and the mastery test. The teacher records on the form information about the student's reading, spelling and typing skills.
7. A sample Teacher Record Form to be reproduced as needed. The teacher records on the form the dates the students pass their tests.
8. A keyboard to reproduce for the student practice at his/her desk on the computer keys.
9. Pretests and post tests for typing and the use of the computer.

Hardware requirements: Apple II, IIe or IIc; one disk drive with 3.3 DOS; a colored monitor is effective but not required.

Cost of the complete program in the binder is $198. Backup disks for additional computers are $60 for each set of four. (Published by Cove Publishers and Educational Priority Systems.)

BLTM has been pilot tested at the elementary and secondary grade levels in the Salt Lake City School District. Students and teachers have been enthusiastic in their praise of the program. A three year study teaching typing and reading skills to Title I first grade students in Salt Lake City

School District showed a 20 percentile average gain in reading as measured by the California Achievement Test. The control group showed a six percentile average gain. Students in grade 3 using the microcomputer program demonstrated reading comprehension and speed-and-accuracy scores, as measured by the Gates-MacGinitie Reading Tests, that are higher than scores of students in a non-equivalent control group, at a statistically significant level ($p < .001$).

BLTM does not replace the teacher introducing the keyboard, letter names, sounds, new words and comprehension skills. It provides opportunities for students to master the skills through enthusiastic and reinforced practice.

ABOUT THE AUTHORS:

Marvis F. Collett, M.S., is a special education teacher (Resource Coordinator), Salt Lake City School District, with certification in Elementary Education and Learning Disabilities. She is the Computer Specialist for the Special Education Department of the District, Computer Specialist for the regular education program of the District (part-time), chairman of Utah Special Education Committee for Computer Technology, member of State Advisory Committee for Computer Technology and member of the textbook and curriculum evaluation committee for computers for the Utah State Office of Education. Mrs. Collett has taught numerous classes in Computer Literacy as well as in Basic Programming skills for teachers, both regular and special education, has conducted many workshops throughout the state and has authored computer programs that are in use nationally.

E. Charles Parker, Ed.D., administrator and former Utah State Office of Education Specialist in Business and Office Education, received his doctorate in Business Education from the University of California at Los Angeles with an

emphasis in curriculum development and media technology. He has been a teacher of typewriting skills in grades 7-12 and has taught teachers in business education at Utah State University.

Ethna R. Reid, Ph.D., Director of the Exemplary Center for Reading Instruction, is the author of reading and language instruction texts. She has been a secondary and an elementary teacher, elementary principal and supervisor and supervisor of language arts. She directs the Reid Foundation Reading Clinic, teaches Educational Psychology courses at the University of Utah and was a former member of the Board of Directors of the International Reading Association (1975-1978). The Exemplary Center for Reading Instruction is a validated Developer Demonstrator Project with the National Diffusion Network Division of the U.S. Department of Education.

Basic Literacy Through Microcomputers was approved by JDRP in 1984 for students in grades one and three. It is a funded National Diffusion Network Project.

Inservice time of two-days is desirable to assist teachers to learn to teach the use of the computer, reading, spelling and typing skills.

CP/M AND MORE, FOR YOUR APPLE

Ron Richter and R.W. McFarland

The intent of this presentation is to demonstrate the use of the Apple II + as a base for a CP/M operating system as well as the conventional Apple 3.3 DOS. In addition we will demonstrate the use of several pieces of software in a non-integrated format beginning with Think Tank by Living Videotext, going to WordStar with the files created with Think Tank, checking the completed files with the Random House Proofreader and finally printing the files created with a typesetting program entitled Fancy Font from Softworks. THe latter program allows the creation of more than thirty different type fonts on the inexpensive Epson MX, FX or Gemini-10x printers. At this point we will use examples of files created by students for various school publications such as the newspaper, yearbook, etc. each in different fonts including Roman, San Serif, Script, Old English and Fancy Font. The last portion of the program will be a discussion of the hardware installations in the Apple II+ which allows the above implementation of CP/M. The hardware demonstrated will include the PCPI-Applicard, the Vista A-800 8 inch double sided, double density disk drives, the Ultraterm from Videx, a special Whisper buffered printer interface and the Epson FX printer.

Outline - Summary of Wordstar, Fancy Font, and Proofreader Programs in Language Arts Classrooms

This presentation will concentrate on the specific uses of the Wordstar word processing program, the Fancy Font printing program, and the Proofreader spelling editor with particular emphasis on language arts and journalism classes. Those in attendance will see how Wordstar saves countless hours of work during editing, laying out, and justifying news stories. This program also facilitates formatting a variety of other writing forms. The print capabilities of Fancy Font will also be demonstrated particularly in the area of journalism. In addition to the word processing program and print program, a demonstration of classroom uses of the proofreader spelling editor will be presented.

I. Newspaper Utilites

- A. Column justification
- B. Editing capabilities
- C. Type selection with Fancy Font
- D. Creating advertisements and announcements
- E. Storing files

II. General Class Utilization

- A. Printing students' writings
- B. Printing programs and announcements
- C. Student editing of their own work
- D. Storing often used modes of writing
- E. Creating a variety of tests, quizzes and assignments

Think Tank Presentation

The purpose of this presentation is to introduce the Think Tank software package as a learning tool for students to structure their thought processes. It will also endeavor to show how to utilize the Apple computer as a base for manipulating files generated with Apple DOS and convert them to the C/PM operating system.

The first segment will cover booting up and operating the Think Tank software as an outline generator.

The secord segment will cover storage and printing files within Think Tank.

The last segment will cover the transfer of Think Tank files from DOS to Pascal to C/PM through the Translate utility of Think Tank.

First Segment

1. Boot up Think Tank Program disk.
2. COMMAND level menu and operation.
3. Changing SUMMIT message using EDIT - HEADLINE
4. Adding new paragraph using NEW - DOWN
5. Additional menus

Second Segment

1. Editing a paragraph using EDIT - PARAGRAPH
2. Ending EDIT feature using ESC - SAVE
3. Printing a file using PORT - PRINTER - FORMATTED
4. Checking format settings and saving
5. Printing

Third Segent

1. Using menus to TRANSLATE
2. Use Up arrow key to get to top of outline
3. From Top Level use PORT - TEXTILE - SEND - STRUCTURED
4. At request for name of Pascal textile, enter #4:Name
5. Think Tank will ask "REMOVING THE THINK TANK DISK?
6. Replace program disk with Pascal formated disk, Answer - Y
7. DONE! Replace Think Tank Pascal disk with Program disk

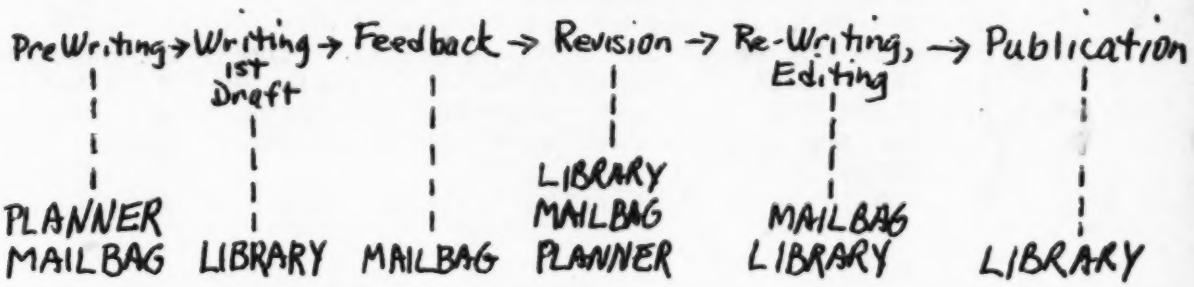
Writing with Quill

Marcia Romick

The Quill program has been commercially available since fall of 1984 and Alaskan teachers have had some time to use and evaluate the program. This session will demonstrate the Quill program and describe some of the applications within the state in various types of classrooms at various grade levels. A further description of the dissemination year of Quill within Alaska, written by Carol Barnhardt, can be found within the Proceedings for last year's Computer Conference.

The first point to be made about Quill is that it is NOT a word processing program; while a word processor is available within the segments of Quill, the purpose of the program is to set up a writing environment for students which promotes writing via a process approach. Quill was not designed as a word processor for teachers but as a writing tool for their students. Because of its process approach Quill supports the Alaska Writing Project and serves as an extension of the Writing Project into computers.

Below is a chart which shows the sections of Quill and the specific parts of the writing process which they enhance.



Prewriting is a step often slighted in the act of writing in the classroom; we somehow expect students to come up with a full-blown product without time to think about what should be included, what sequence the topics might take or the other preliminaries that no professional writer would omit whether they went onto paper or were generated in the writer's head. PLANNER, in Quill, allows the teacher or student to jot down notes, as a reminder of what should be answered or covered in a piece of writing.

MAILBAG is the communication section of Quill, set up to ease the quick reading and writing of messages to classmates. While it is possible to edit these messages in a limited way, this section is intended for spontaneous, easy communication between youngsters and teachers; elaborate revision and editing of writing takes place in LIBRARY.

LIBRARY, the main section of Quill, has a full word processor, WRITER'S ASSISTANT. While the most elaborate functions of word processing can be carried out here, only a few commands need be learned at a time so that fairly young children can learn to handle it. Writing can be cataloged by means of key words, forcing students to focus on main ideas. The cataloging system allows other readers to look at the writing for feedback or to consult the classroom experts on a subject just as they would check the card catalog in a school library.

The formatting commands will list the sentences in a paragraph under each other making it visually obvious if such

things as beginning capitalization and ending punctuation are there. More experienced writers can look for varying length and structure of sentences and whether all sentences start in the same way.

With its unified commands and format throughout the three sections, Quill is easy for students to learn. In a resource room situation, the teacher reports that her L. D. students preferred Quill to Bank St. Writer because they found it less confusing. (The teacher liked it because it involved quite a bit of reading.)

No computer program will solve all learning problems in all situations. Quill is no exception to that rule--but it is worth a close look.

Marcia Romick
Fairbanks NSB School
District

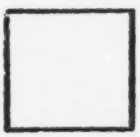
APPLE LOGO PROJECTS FOR TEACHERS AND CHILDREN*

by
Donna Gail Shaw
University of Alaska, Anchorage

Most of the projects done with children and Logo involve turtle graphics. Some of the more popular beginning projects involve the creation of geometric shapes such as squares, triangles, and circles.

```
TO SQUARE

FD 50
RT 90
FD 50
RT 90
FD 50
RT 90
FD 50
RT 90
HT
END
```



The REPEAT command can be used to shorten this procedure.

```
TO NEWSQUARE
REPEAT 4 [FD 50 RT 90]
END
```

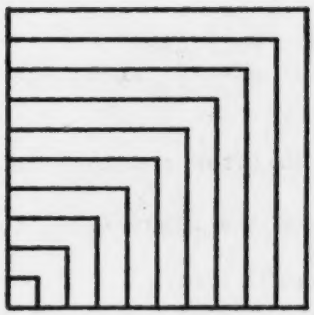
The addition of a variable to this procedure can make it more fun.

```
TO VARSQUARE  :SIDE
REPEAT 4 [FD  :SIDE RT 90]
END
```

Each time you type VARSQUARE you must give it a number. Logo will take this number and use it as the number of turtle steps the turtle will move in the forward direction. The following sequence of directions will produce an interesting design:

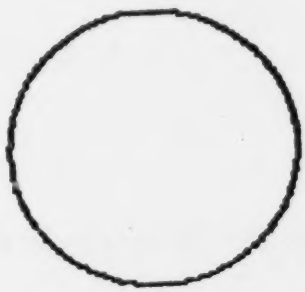
*Portions of this paper will appear in Computers in Early Childhood Education: Issues and Practices, edited by James L. Hoot, published by Prentice-Hall.

```
VARSQUARE 10
VARSQUARE 20
VARSQUARE 30
VARSQUARE 40
VARSQUARE 50
VARSQUARE 60
VARSQUARE 70
VARSQUARE 80
VARSQUARE 90
VARSQUARE 100
```



Experimenting with circle procedures can be fun.

```
TO CIRCLE
REPEAT 360 [FD 1 RT 1]
END
```



Varying sizes of circles can be made by changing the 360 and the forward and right inputs. For example:

```
TO NEWCIRCLE
REPEAT 180 [FD 1 RT 2]
END
```

This new circle procedure draws a smaller circle.

A more challenging and exciting procedure is one that can produce polygons through the use of recursion. Recursion is the ability of a procedure, such as MAKEPOLY, on the following page to use the term MAKEPOLY as part of its definition. Generally, it is the ability of the procedure to call itself.

```
TO MAKEPOLY  :SIDE  :DEGREE
FD  :SIDE
RT  :DEGREE
MAKEPOLY  :SIDE  :DEGREE
END
```

To use this procedure, you must enter two numbers (side and degrees) when you type the procedure name. Since this procedure calls itself, it will run indefinitely. To stop the procedure, type CONTROL-G.

MAKEPOLY 80 144

MAKEPOLY 50 120

MAKEPOLY 50 90

AIDS FOR BEGINNERS

The Apple version of Logo contains a file on the Logo diskette known as STARTUP AIDS. It is loaded into the workspace when Logo starts up. Six arc procedures and two circle procedures are in the STARTUP file when you buy the Logo disk. These procedures are simple to use.

1. ARCLEFT and ARCL take two inputs--radius and degrees. It draws an arc of the designated number of degrees curving toward the left, taken from a circle with the designated radius.
2. ARCRIGHT and ARCR--same as above but curves toward the right.

For example, ARCR 25 90 will draw a quarter of a circle with a radius of 25 turtle steps.

3. CIRCLEL takes one input--radius. It draws a circle of the designated radius curving to the left.

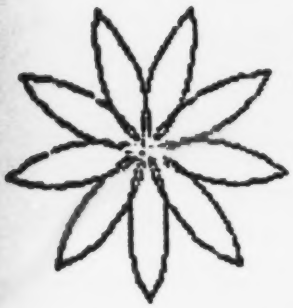
4. CIRCLER--same as above but curves to the right.

For example, CIRCLER 40 will draw a circle with a radius of 40 turtle steps.

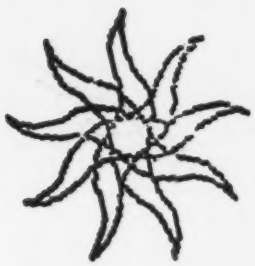
155

Some very interesting and complex designs can be made using the aids with very little difficulty on the part of the programmer. For example:

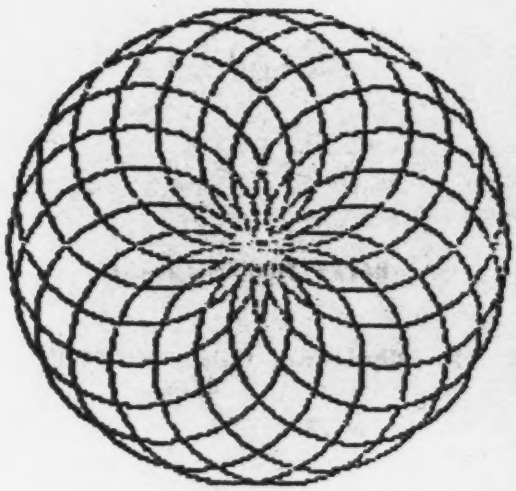
(see attached listing - Aids for Beginners)



FLOWER



FIREBALL



FANCYCIRCLE

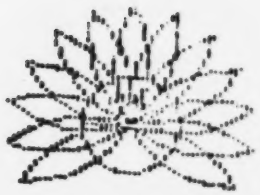
Seasonal Projects

For holiday activities, some teachers enjoy programming seasonal projects and allowing their students to modify them or view them as motivators for developing their own seasonal designs. The following are teacher-created designs. Most of them make use of the Logo STARTUP AIDS found on the Logo disk.

The SETCURSOR command is used to print the messages found at the bottom of the screen on each seasonal project. SETCURSOR takes two numbers which designate the column and row to which the cursor is to be moved. Columns are numbered 0 through 39. Rows are numbered 0 through 23. For example, SETCURSOR [20 22] would put the cursor in column 20 and row 22. Subsequent PRINT commands would begin printing from this position.

See attached listing--

1. Thanksgiving (TURKEY)



HAPPY THANKSGIVING!!

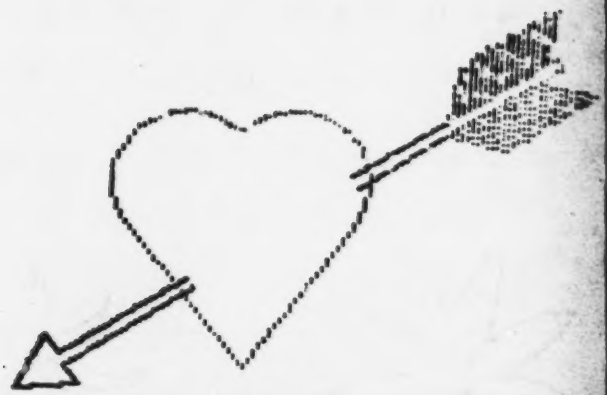
2. Christmas (CHRISTMASTREE)

MERRY CHRISTMAS!!

3. *SNOWMAN

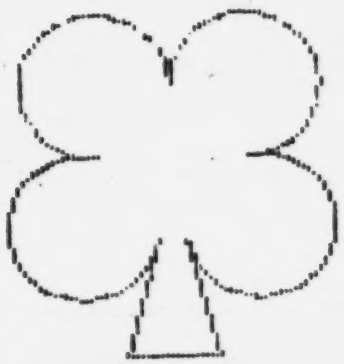
FROSTY THE SNOWMAN

4. *Valentine's Day (VALENTINE)



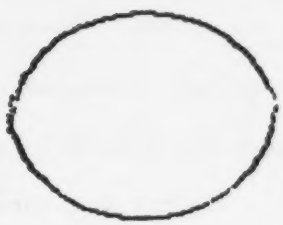
BE MY VALENTINE

5. *St. Patrick's Day (CLOVER)



HAPPY ST. PATRICK'S DAY

6. Easter (EASTEREGG)



HAPPY EASTER EGG HUNTING!!

When drawing on a colored background (2 - 5) only two of the four colors (green, violet, blue, orange) are available. When the background is green or violet, blue and orange cannot be used (pencolor 4 draws in green and pencolor 5 draws in violet). When the background is blue or orange, pencolor 2 draws in orange and pencolor 3 draws in blue. Strange effects may occur any time you try to mix colors. A good example of this is the EASTEREGG program in which only the background color is changed, however, when executed, it appears that the pencolor has been changed also.

*Courtesy of Becky Northam, Lynne Matcek and San Jacinto Elementary, Deer Park, Texas

THE BEEPING TURTLE

It is a simple matter to make your turtle beep. The following procedure will produce one beeping sound:

```
TO BEEP
PR CHAR 7
END
```

EASYDRAW for Young Children

The following procedures allow you to move the turtle with single keystrokes.

```
TO DRAW
MAKE "KEY READCHAR
IF  :KEY = "F [FD 5]
IF  :KEY = "R [RT 10]
IF  :KEY = "L [LT 10]
IF  :KEY = "B [BK 5]
IF  :KEY = "C [CS]
IF  :KEY = "D [PD]
IF  :KEY = "U [PU]
IF  :KEY = "E [PE]
IF  :KEY = "Q [STOP]
DRAW
END

TO EASYDRAW
CS
DRAW
END
```

a) To make the turtle move forward, simply press the F key and the turtle will take 5 turtle steps forward each time you press it.

b) To move backward, press the B key and the turtle will move back 5 steps each time you press it.

c) To turn to the right, press the R key and the turtle will turn 10 degrees to the right each time you press it.

d) To turn to the left, press the L key and the turtle will turn 10 degrees to the left each time you press it.

e) To make the turtle stop drawing (PENUP), press U.

f) To make the turtle begin drawing again (PENDOWN), press D.

g) To make the turtle erase (PENERASE), press E.

h) To clear the screen, press the C key.

i) To quit, press the Q key.

Designs created by regular Logo and EASYDRAW may be printed on paper. It will be necessary to save the design as a binary file to be printed. Do the following to save your picture as a binary file:

1) Insert an initialized disk into the drive (NOT the Logo diskette).

2) Type:

.PRINTER 6 (the disk drive will come on and your design will disappear from the screen)

3) Type:

BSAVE picture name, A$2000,L$2000
(NO quote marks)

(Picture name is any name you choose to give your picture.)

Various printers have different procedures for printing out graphics, so it will be necessary to check the printer manual for specific directions for printing out the picture you have stored in the binary file.

Resources

1. Abelson, H. Logo for the Apple II. New Hampshire: BYTE/McGraw-Hill, 1982.

2. Gorman, H., Jr., & Bourne, L. E., Jr. Learning to think by learning Logo: Rule learning in third grade computer programmers. Bulletin of Psychonomic Society, 1983, 21, 165-167.

3. Harvey, B. Why Logo? Byte, 1982, 7 (8), 163-193.

4. Hines, S. N. Computer programming abilities of five-year-old children. Educational Computer, 1983, 3, 10-12.

5. Lough, T. Logo: Discovery learning with the classroom's newest pet. Electronic Learning, March 1983, pp. 49-53.

6. Papert, S. Mindstorms: Children, computers and powerful ideas. New York: Basic Books, Inc., 1980.

7. Poirot, J., & Adams, R. C. Forty easy steps to programming in BASIC and Logo. Austin: Sterling Swift, 1983.

8. Solomon, C. Introducing Logo to children. Byte, 1982, 7 (8), 196-208.

9. Torgerson, S. Logo in the classroom. Oregon: ICCE Publications, 1984.

10. Watt, D. Logo in the schools. Byte, 1982, 7 (8), 116-134.

AIDS FOR BEGINNERS

#1 FLOWER

```
TO PETAL
REPEAT 2 [ARCR 60 60 RT 120]
END

TO FLOWER
REPEAT 9 [PETAL RT 40]
END
```

#2 FIREBALL

```
TO FLAME
REPEAT 2 [ARCL 25 60 ARCR 25 60]
END

TO FIREBALL
REPEAT 9 [FLAME RT 160]
END
```

#3 FANCYCIRCLE

```
TO FANCYCIRCLE
FULLSCREEN
REPEAT 18 [CIRCLEL 50 RT 20]
END
```

#1 TURKEY

```
TO FEATHER
COLOR
REPEAT 2 [ARCR 50 60 RT 120]
LT 20
END

TO COLOR
MAKE "C :C + 1
IF :C = 6 [MAKE "C 2]
SETPC :C
END

TO BODY
PU
SETPC 4
HOME
SETX -20
PD
CIRCLER 20
HOME
END

TO TAIL
MAKE "C 1
RT 90
REPEAT 13 [FEATHER]
END

TO TURKEY
SETBG 1
TAIL
BODY
SETBG 0
SETCURSOR [10 20]
PR [HAPPY THANKSGIVING!!]
END
```

#2 CHRISTMASTREE

```
TO LTREE
CS PU FD 80 PD LT 140 REPEAT 5 [FD 30 LT 120 FD 10 RT 120]
END


TO RTREE
PU
HOME
FD 80 RT 140
PD
REPEAT 5 [FD 30 RT 120 FD 10 LT 120]
SETH 270
FD 40
SETPC 5
LT 90 FD 10 RT 90 FD 15 RT 90 FD 10
LT 90
SETPC 2
FD 42
END
```

```
TO STAR
PD
HT
MAKE "C :C + 1
SETPC :C
REPEAT 5 [FD 20 RT 144]
IF :C = 5 [STOP]
STAR
END


TO ORNAMENTS
SETPC 1
PU HOME PD
CIRCLER 3
PU FD 35 LT 20 FD 15 PD
CIRCLER 3
PU RT 90 FD 25 PD
CIRCLER 3
PU HOME
SETPC 2
RT 180 FD 30 LT 90 FD 10 PD
CIRCLER 3
PU RT 2 FD 30 PD
CIRCLER 3
PU HOME LT 45 FD 23 PD
SETPC 3
CIRCLER 3
PU LT 78 FD 25 PD
CIRCLER 3
PU HOME LT 120 FD 30 PD
SETPC 4
CIRCLER 3
PU LT 45 FD 35 PD
CIRCLER 3
END


TO CHRISTMASTREE
HT
SETPC 2
LTREE
RTREE
ORNAMENTS
PU HOME FD 80 LT 50 BK 8
MAKE "C 0
STAR
SETCURSOR [11 20]
PR [MERRY CHRISTMAS!!]
END
```

#3 SNOWMAN

```
TO SCARF
LT 100
FD 10
LT 90
PD
FD 25 RT 90 FD 1 RT 90
FD 25 LT 90 FD 1 LT 90
FD 25 RT 90 FD 1 RT 90
FD 25 LT 90 FD 1 LT 90
FD 25
RT 45
FD 30
RT 90
FD 1
RT 90
FD 29
LT 90
FD 1
LT 90
FD 28
RT 90
FD 1
RT 90
FD 29
LT 90
FD 1
LT 90
FD 29
BK 31
RT 35
FD 30
LT 90
FD 1
LT 90
FD 30
RT 90
FD 1
RT 90
FD 30
LT 90
FD 1
LT 90
FD 30
HT
END

TO PIPE
LT 90
FD 38
RT 100
PD
FD 10
REPEAT 4 [FD 3 RT 90]
PU
END
```

```
TO HAT
FD 32
LT 90
PD
FD 25
BK 35
FD 10
RT 90
FD 10
LT 90
FD 15
LT 90
FD 10
RT 90
BK 5
PU
END


TO FACE
FD 80
LT 90
FD 10
PD
CIRCLEL 3
PU
BK 20
PD
CIRCLEL 3
PU
FD 10
LT 90
FD 10
RT 90
BK 3
PD
REPEAT 3 [FD 6 RT 120]
PU
FD 3
LT 90
FD 10
RT 90
PD
ARCR 7 90
RT 180
ARCL 7 90
ARCL 7 90
PU
END
```

```
TO BUTTONS
PD
CIRCLEL 5
LT 90
PU
FD 20
RT 90
PD
CIRCLEL 5
LT 90
PU
FD 20
RT 90
PD
CIRCLEL 5
RT 90
PU
END
```

```
TO BODY
LT 90
CIRCLEL 35
CIRCLER 25
LT 90
PU
FD 10
RT 90
END
```

```
TO SNOWMAN
SETPC 1
BODY
SETPC 3
BUTTONS
SETPC 1
FACE
SETPC 5
HAT
PIPE
SETPC 4
SCARF
SETCURSOR [11 21]
PR [FROSTY THE SNOWMAN]
END
```

```
#4 VALENTINE
TO FEATHERS
SETPC 5
REPEAT 15 [FD 25 BK 25 LT 45 FD 3 RT 45]
LT 45
FD 10
LT 90
FD 5
LT 90
FD 50
RT 135
REPEAT 16 [FD 25 BK 25 RT 45 FD 3 LT 45]
HT
END

TO ARROW
PU
SETPC 1
RT 45
FD 30
LT 110
FD 90
RT 170
PD
FD 45
PU
FD 80
PD
FD 60
RT 90
FD 10
LT 120
FD 25
LT 120
FD 25
LT 120
FD 10
RT 90
FD 60
PU
FD 80
PD
FD 40
RT 45
END

TO LVALVE
RT 105
FD 75
ARCR 30 170
END
```

167

```
TO RVALVE
CS
CLEARTEXT
PU
SETPOS [-10 10]
SETPC 4
PU
FD 40
RT 45
PD
ARCR 30 170
FD 75
END


TO VALENTINE
RVALVE
LVALVE
ARROW
FEATHERS
SETCURSOR [13 20]
PR [BE MY VALENTINE]
END
```

```
#5  CLOVER
TO STEM
FD 50
BK 50
PU
SETPOS [-15 0]
PD
RT 30
FD 50
SETH 90
FD 35
HT
END


TO RCLOVER
REPEAT 2 [ARCR 30 270 RT 170]
RT 5
END


TO LCLOVER
SETPC 2
PU
SETPOS [-15 0]
LT 170
PD
REPEAT 2 [ARCR 30 270 RT 170]
RT 10
END


TO CLOVER
CS
CLEARTEXT
LCLOVER
RCLOVER
STEM
SETCURSOR [6 20]
PR [HAPPY ST. PATRICK'S DAY!!]
END
```

```
#6  EASTEREGG
TO PAINT
HT
SETBG 0
SETCURSOR [18 21]
PR "BLACK
WAIT 60
SETBG 1
SETCURSOR [18 21]
PR "WHITE
WAIT 60
SETBG 2
SETCURSOR [18 21]
PR "GREEN
WAIT 60
SETBG 3
SETCURSOR [18 21]
PR "VIOLET
WAIT 60
SETBG 4
SETCURSOR [18 21]
PR "ORANGE
WAIT 60
SETBG 5
SETCURSOR [18 21]
PR "BLUE
END

TO EGG
CS
PU
SETPOS [-50 0]
PD
ARCR 60 170
RT 15
ARCR 60 170
END

TO EASTEREGG
EGG
PAINT
SETCURSOR [8 23]
PR [HAPPY EASTER EGG HUNTING!!]
END
```

What Should Students Be Learning With Computers?

Nancy Parks Sopp
Computer Lab Teacher
Fairbanks North Star Borough School District

Not even the experts agree about what students should be doing with computers. Dr. Arthur Luehrman, the educator who coined the term "computer literacy," recommends that students take a semester-long course that emphasizes programming skills. The editor-in-chief of The Computing Teacher, Dr. David Moursund, recommends that students use the computer in many subject areas, throughout their school career.

Research studies don't offer much help, either. Some evidence indicates that computer-assisted instruction in math boosts achievement scores, but you can't base a computer curriculum on math drill.

The workplace will continue to change as a result of computer technology. But just what skills will students need when they graduate from high school? I decided to pose the question to community members.

I interviewed 16 community members, whose professions ranged from professor to independent contractor, whose computer knowledge ranged from fluent Fortran to "I'm thinking of getting a computer." Most of them hire, supervise or teach young adults.

© 1985 by Nancy Parks Sopp

I did not interview any computer programmers or systems analysts. The number of jobs in the computer field is growing rapidly; nevertheless, most young people will choose other vocations.

Do students need to learn to use a computer? The answer was an unequivocal "yes," even though computer skills are not yet required for many jobs.

The owner of the local Wendy's restaurant franchise, Paul Frith, said, "Somewhere in the curriculum, kids have got to learn that computers are part of life."

"In the future, anyone who can't use a computer will be lost," said Jeanne Rogers, a vice-president for First National Bank of Fairbanks.

But just what should they learn? Everyone agreed that students should understand the basics of how a computer works, but not necessarily the details. "They don't have to know how a 6502 microprocessor works," said Bruce Wammack, a real estate broker and borough assemblyman.

Norm Kolb, manager of Yukon Office Supply, said, "Young people should learn that there's nothing magical about the computer. They should gain a basic understanding of programming, to learn how a computer works. But they should move on to practical applications as soon as possible." Kolb has computerized some, but not all, of his business operations, finding that doing some things the old way gives more accurate results.

The president of Market Basket, Inc., Dave Kilbourn, said, "Kids should learn that the computer is a tool, not a solution. And the most important tool an individual can possess is an inquiring mind." Kilbourn deplored the scarcity of this organic tool: "Problem-solvers are few and far between," Kilbourn said.

Kilbourn's perception is reinforced by the findings of The National Assessment of Educational Progress, whose research results indicate "little evidence of well-developed problem-solving strategies or critical-thinking skills" among youngsters.[1]

John Raich, an engineer for Golden Valley Electrical Association, believes that problem-solving--not computer knowledge--is the most vital skill for young people. Raich frequently works with computers on his job, and even does some programming. "But all the programming knowledge in the world won't do you any good if you can't think. You've got to be able to determine the steps involved in finding a solution--and then judge if your answer is correct. If you can think logically, you can easily pick up whatever computer skills you need. If you lack the thinking skills, you won't get far with computers."

The National Council of Teachers of Mathematics recognizes that logic is paramount, recommending, in its An Agenda for Action, that the school mathematics curriculum focus on problem-solving throughout this decade.[2] In a companion text, An Agenda In Action, problem-solving

activities in math are described for all grade levels, some of which use the computer.[3]

Former state legislator Brian Rogers believes that the computer is an excellent tool for teaching logic, by means of a programming language or a spreadsheet. Rogers regards programming as a tool for teaching logic, not as a vocational skill. "It's more valuable to understand how a spreadsheet works than to be able to program in Fortran," he said. Rogers uses applications software--spreadsheets, word processors, and graphing programs--extensively in his work as budget director for the University of Alaska, Fairbanks.

In some professions, programming skill is required. Dr. L. Claron Hoskins, Head of Chemistry at the University of Alaska, Fairbanks, said, "The most important thing for students to know is how to turn a computer on and write simple programs in BASIC." This understanding of how a computer works will prepare chemistry students to learn Fortran. His sentiments were echoed by Dr. Robert Carlson, Head of Civil Engineering at the University of Alaska, Fairbanks, who wants to change the entrance requirements for freshmen in engineering to include 1/2 unit of computer programming.

An education professor at the University of Alaska, Fairbanks, Dr. Dick Riedl, provides a broader perspective on the topic. "I would like to see schools stress use of applications software to learn the curriculum," he said.

"Teachers should provide students with a real use for the computer."

A very practical use of computer technology is to use it to teach vital economics concepts. Bruce Wammack said, "I interview young people who can program games, but don't understand how a checking account operates." Wammack suggests that students use a spreadsheet to learn economics concepts such as personal budget planning, how and why interest is paid, and how to determine the return on an investment. In addition to learning financial modeling, Wammack recommends that students learn keyboarding and writing skills.

Paul Frith agreed that keyboarding skills are very important. He proposed a "computer appreciation" course, integrated into "someone's" curriculum. By high school, students would benefit from a computer applications unit. Frith uses a spreadsheet to manage the finances of his business ventures.

From my small survey, I found no pat answer to my question, "What should students be learning with computers?" One characteristic that all those I interviewed shared is that they continually ask questions. They use computers--in various ways--only when computers help in finding the answers.

Come to think of it, maybe I found the answer to my question after all.

---

[1] Sanoff, A.P. & Solorzano, L., "It's at Home Where our Language is in Distress," U.S. News & World Report, Feb. 18, 1985, p. 54

[2] An Agenda for Action, National Council of Teachers of Mathematics, c. 1980.

[3] An Agenda in Action, National Council of Teachers of Mathematics, c. 1980.

Ron Thorkildsen, Ph.D.
Director of Technology and Administrative Services

Associate Professor in Special Education
and Instructional Technology

USING A MICROCOMPUTER CONTROLLED VIDEODISC PLAYER IN THE EDUCATION OF MENTALLY HANDICAPPED STUDENTS

With the evolution of the computer, the promise of Computer Assisted Instruction (CAI) to provide a vehicle for individualizing instruction became more and more apparent. However, a major problem exists when trying to use CAI with the mentally handicapped. Traditional CAI and individualized, self administered assessment instruments require the student to have reading skills. This essentially precluded a majority of the population of handicapped students. Consequently, it is necessary to use spoken instructions to interact with the student. Until very recently the technology has not been available to do this. However, with the advent of microelectronics and laser optics, the technology is now available. Recently developed videodisc players coupled with the microcomputer provide the technology to deliver the spoken instructions and questions necessary to communicate with a non-reader.

Videodisc Technology

Videodisc technology consists of the videodisc, the videodisc player, and the equipment necessary to produce the videodisc.

The videodisc provides for a high volume of storage of both audio and video information. It is capable of storing 54,000 individual frames per side. Each frame can contain a single picture, such as a 35mm slide image. When the single frames are played at 30 frames per second, each side of the disc contains 30 minutes of motion. Associated with the single video track are two audio tracks. The audio tracks are independent and may be used in a variety of applications, such as recording two different languages associated with the same video. The videodisc is truely an omnibus medium; it has the capability of presenting motion film, slides, graphs, and high quality audio.

The videodisc player may be played at regular speed, in single-frame stop motion, and at various speeds of slow motion. All of the functions can be selected easily, but the most impressive feature of the videodisc is rapid random-access capabilities. It searches all 54,000 frames in approximately three seconds. In general, search times in educational applications take one to two seconds. For example, a segment of instruction can be presented, and depending on the learner's response, the videodisc can present a video and audio feedback segment appropriate to the learner's response within one second.

Laser optic videodisc systems are being produced by a number of organizations including Hitachi, SONY, Thompson, Pioneer, and Magnavox. The Pioneer LD 700 and Magnavox players are designed primarily for home entertainment use, while the Pioneer LD-V4000 and LD-V1000, Thompson, Hitachi, and SONY are designed for

educational and industrial use. Rapid random access is the primary advantage of the education/industrial models. Entertainment models sell for approximately one-third the cost of the educational/industrial models.

Controlling the Videodisc Player

There are a number of options for controlling or operating a videodisc player. They are typically referred to as levels of interactivity and are described below:

Level 1 - The videodisc player is controlled manually with a remote-control device. This device has function keys for each of the player's options. For example, to search for a particular frame on the disc, the operator enters the frame number on the remote-control device and pushes the search key to initiate the search. After finding the desired frame, the operator has numerous options such as forward or reverse play, single-frame display, slow-motion play or regular play with or without audio from either or both of the audio tracks. All of these functions are accessible from the remote-control device.

Level 2 - Some educational/industrial models such as the SONY and the Pioneer 7820 and LD-V6000 have built-in microprocessors. All functions mentioned above can be controlled by this microprocessor. The computer program that determines the control is stored on the videodisc and then loaded from the disc into the microprocessor. The flow of the presentation is controlled by the logic in the computer program and by input from a user with the remote-control device.

Level 3 - The videodisc player is interfaced with an external computer. The logic of the presentation is determined by the computer program in the external computer, and by input from the user, usually entered through the computer's keyboard. This is advantageous because the computer can supply text and graphics in addition to the still frames and motion supplied by the videodisc. Additionally, student progess data can be collected and stored on an external storage device such as a floppy disc. Typically, microcomputers are used as the external computer, and videodisc interfacing is now possible with all popular personal computers.

The IVSET Program

Approximately six years ago, the Interactive Videodisc for Special Education Technology (IVSET) program was started at the Exceptional Child Center. The major purpose of the IVSET program is to develop and field test computer aided instruction and assessment systems for mentally handicapped students. Through the efforts of the IVSET program staff, hardware, software, and instructional techniques have been developed which provide both computer assisted instruction and computer aided individualized assessment for mentally handicapped students.

Five seperately funded projects have been conducted under the auspices of the IVSET program. Two of these projects, the Basic Skills Project and the Social Skills Project, are described in this paper.

Basic Skills Project

A major objective of the Basic Skills Project was to develop a microcomputer-videodisc (MCVD) system. The hardware for the MCVD system consists of a Pioneer Model 7820-III Videodisc player, an Apple II Plus microcomputer with two 5 1/4" floppy disc drives, a SONY 12" color television monitor and a Carroll Mgf. touch panel built into the television screen. The videodisc player was selected for its rapid, random access capabilities and excellent still frame quality. (A typical search and retrieval of an instruction or feedback segment takes less than 1 second). The Apple II controls the system through computer programs and an interface device (Allen VMI Board) designed for use in the Apple. Both were designed and developed by IVSET Program staff. The videodisc is the storage medium. It has the approximate size and appearance of an LP phonograph record and is capable of storing 54,000 individual frames of video or 30 minutes of audio and motion video on each side. It also has two independent audio tracks.

Students interact with the system by viewing visual material on the television screen and by responding to spoken questions or instructions. The student responds by touching the appropriate image of an object on the television screen. The location of the television screen touched by the student is detected by the touch panel. This location is then transmitted to the computer. The computer program in the microcomputer contains the correct location for each segment of instruction.

The location transmitted by the touch panel is compared to the correct location to determine if the student's response is correct or incorrect.

On a correct response, the microcomputer responds by finding and retrieving a segment on the videodisc which contains positive feedback. Other possible response conditions are an incorrect response, and non-response. Segments are presented on the videodisc for these response conditions as well as a variety of feedback, including animation and motion picture sequences.

Each segment of instruction has associated parameters that specify the number of times a student must respond correctly to advance to the next instruction segment. As the student interacts with the system, data are collected by the microcomputer and stored on a floppy disc by the Apple disc drive. Six instructional programs were developed for use with the MCVD system during the first project: (1) Matching Sizes, Shapes and Colors, (2) Time Telling, (3) Identification of Coins, (4) Functional Words, (5) Sight Reading, and (6) Directional Prepositions. Each of these programs were field tested during the course of the project.

The major goal of all field tests was to obtain information necessary for the continued development and refinement of the system. The following populations were involved in the field testing.

a. Elderly mentally retarded - Brigham City Day Care Center

b. Young (4-13 yrs) - Severe to moderately mentally retarded - Matching - Exceptional Child Center - Matching - Exceptional Child Center - Prepositions.

c. All age severe to moderately mentally retarded
--Utah State Training School
--Identification of Coins
--Functional Words

d. Young (7-10 yrs) - Learning disabled and mildly mentally retarded
--Resource Room - Time Telling
--Resource Room - Beginning Readig

Investigation of different populations was conducted both to determine if the students learned from using the system and if the student can work independently with the system. It is a major goal to have the operation of the system as independent of the teacher as possible.

Because of the formative nature of the evaluations, and the resulting research designs, generalization beyond the actual examples has not been made. However, based on pre-test and post test gains, we can conclude that both statistically and practically (educationally) significant learning was evident with the following programs:

1.	Matching	Brigham City
2.	Identification of Coins	Training School
3.	Functional Words	Training School
4.	Time Telling	Logan Res. Room
5.	Beginning Reading	Logan Res. Room

The data for each of these projects is published in the final report of the project.

The major purpose of the Basic Skills Project was to investigate the feasibility of using this new technology with various programs and with students with varying degrees of learning difficulties. None of the instructional programs are in a complete form because it wasn't necessary to have a complete program to conduct the research, nor were there sufficient funds available to complete the program. However sufficient evidence was collected to suggest that the MCVD system was very strong potential for providing effective CAI for handicapped students.

The Social Skills Project

The Social Skills Project combined both videodisc material and written material to teach social skills to fourth, fifth and sixth grade emotionally disturbed children. The resulting instructional program is referred to as the Interactive Videodisc Social Skills (IVSS) Program.

The IVSS program teaches children how to use appropriate phrasing, intonation, and body language in such social interactions as getting involved and being positive. The videodisc is used to present (1) examples of appropriate and inappropriate social behaviors, and (2) models to imitate in subsequent role playing activities. A daily lesson guide for the teacher accompanies each videodisc presentation. The program also includes a behavior management system that is used

during and after the videodisc and role playing phase.

A research study was conducted from January through May of 1984 to investigate the effectiveness of the Interactive Videodisc Social Skills (IVSS) program for children who have been identified as mildly handicapped and who have been placed in a resource room for part of their daily instruction.

The investigation determined whether social skills training can increase the number of positive social interactions of handicapped children toward their non-handicapped peers, reduce the incidence of negative social behaviors, improve their self-concept, and increase acceptance by their non-handicapped peers.

Six elementary school resource rooms, each containing five mildly handicapped students, were randomly assigned to participate in the program (experimental group) or to continue their regular resource room program (control group). The students were classified as neglected, accepted, or rejected. Data on the student's social behavior, acceptance by nonhandicapped peers, self-esteem, and treatment implementation were collected over a four month period.

The students in the experimental group scored significantly higher on a post-training measure of peer acceptance than did control group students. Also, within the experimental group the neglected and accepted students scored higher than did rejected students. The experimental and control group students did not differ on the post-measure of self-esteem.

Experimental group students, irrespective of their classification, were rated significantly higher than control group students on a post-checklist of social skills covered in the program. This checklist was completed by each students' resource room teacher.

From this study it was concluded that the experimental group students learned the social skills taught by the program, their positive behaviors were increased, and peer acceptance by their regular classroom peers was significantly improvements in peer acceptance.

A complete description of the data analysis is contained in the final report of this project. This report is available from the author.

DIFFERENCES BETWEEN MALE AND FEMALE STUDENT INVOLVEMENT WITH COMPUTERS IN SCHOOLS

Ralph Van Dusseldorp
Professor of Education
University of Alaska
Anchorage, Alaska 99504

Norman Holthouse
Director, Management Information Systems
Anchorage School District
Anchorage, Alaska 99502

Introduction

For a number of years there has been a national awareness, and a concern on the part of many, that males are more likely than females to enter careers involving mathematics and science. Studies have shown that boys are not only more likely than girls to enter mathematics and science related careers, they also score higher on mathematics and science achievement and aptitude tests (Allen and Chambers, 1977; Blum and Givant, 1980; Fox, 1976; Pedro et al, 1981; Sells, 1976; Young, 1975).

Most researchers attribute the differences between male and female mathematics and science aptitude, achievement and career choices to social-cultural factors rather than inate ability (Boswell and Katz, 1980; Brush, 1979; Fennema and Sherman, 1977; Frazier and Ladker, 1973; Greenburg, 1978; Henry, 1981; Kominski, 1976; Kirk, 1975).

Now it appears that there is also a gap between males and females in computer skills, thus further limiting females in careers in mathematics, science and technology. It also appears that this computer gap, as is the case of the mathematics and science gaps, may be attributed to social-cultural conditioning (Kiesler, Sprouel and Eccles, 1983; Miller, 1970; Naiman, 1982).

Casual observations in the Anchorage School District indicated a difference between male and female students in computer learning and utilization. The purpose of this study was to determine if such a difference in fact existed, the

extent of the difference if it did exist, the areas in which it existed, and the possible reasons for the difference.

Sources of data for this study included attitude instruments administered to elementary school students, teachers and parents, enrollment in secondary school computer, mathematics and science courses, and participation in extracurricular computer activities.

Attitudes of Students, Teachers and Parents

Attitudinal information was gathered in four elementary schools that were heavily involved with the use of computers. Two of the schools had a computer in each classroom while the other two schools had computer laboratories containing nine and eleven computers respectively. Thus, the students and teachers surveyed had had considerable experience with computers. Attitudinal data was collected in November 1983 and May 1984.

Student Attitudes

There was almost no difference between boys and girls in their attitudes toward computers. Both groups were extremely positive toward computers. Over 92 percent of both groups felt that they liked using computers, that computers are important to have in schools, that computers can help them learn, that they would like to use computers more in school, and that computers are fun. However, there was a significant difference between boys and girls in the availability and use of computers outside of school. Thirty-seven percent of the boys and 28 percent of the girls reported that there was a computer in their home that they could use. Sixty-four percent of the boys and 51 percent of the girls reported that they had used a computer at home or someplace else outside

of school. Thus even though there was no difference in their attitudes toward computers, more boys than girls had access to and used computers outside of school.

Teacher Attitudes

Less than five percent of the teachers reported that the boys in their classes used computers more than the girls. Only one teacher did not feel that girls in the class should use computers as much as boys. Ninety-one percent of the teachers felt that it is important for girls in their classes to learn to use computers compared to 80 percent who felt that it is important for boys in their class to learn to use computers. Comments made by teachers revealed that several felt that it is more important for girls to learn to use computers in school since they are less likely than boys to learn to use computers outside of school.

Parent Attitudes

The attitudes of parents of boys were compared to the attitudes of parents of girls. Comparison of individual attitudinal items revealed only very slight differences between the parents of boys and girls with just as many differences favoring girls as boys. When all attitude items were combined the average response for parents of boys was exactly the same as for parents of girls.

Enrollment in Computer Courses

In order to determine if more male than female students enrolled in high school computer courses, the enrolment in computer courses for 1983-84 and

1984-85 for all Anchorage high schools were examined. Those figures revealed that:

* The enrollment in all high school computer courses consisted of 933 (53.0%) boys and 829 (47.0%) girls.
* The enrollment in the basic Introduction to Computers course consisted of 607 (51.5%) boys and 572 (48.5%) girls.
* The enrollment in the advanced Special Topics in Computers consisted of 268 (72.2%) boys and 103 (27.8%) girls.

Thus the introductory and basic computer courses enrolled more boys than girls but not by a large margin. However, the advanced Special Topics course enrolled over twice as many boys as girls.

Comparisons were also made between enrollment in computer courses and enrollment in high school elective advanced mathematics and science courses. Enrollments in the elective advanced mathematics and science courses for 1983-84 and 1984-85 showed that:

* 3,476 boys (51.1%) and 3,326 girls (48.9%) were enrolled in elective advanced mathematics courses (Algebra II, Geometry, Trigonometry, Calculus, etc.).
* 201 boys (57.1%) and 151 girls (42.9%) were enrolled in the most advanced mathematics course (advanced placement calculus).
* 2,067 boys (50.9% and 1,995 girls (49.1%) were enrolled in elective advanced science courses (Biology II, Chemistry I and II, Physics, etc.).
* 331 boys (50.5%) and 325 girls (49.5%) were enrolled in the most advanced science courses (Advanced Placement Biology II and Advanced Placement Chemistry II).

Comparing the enrollment figures for computer courses with those for mathematics and science courses shows that when all courses in each area are combined, each area enrolls more boys than girls but the difference is not great -

53.0% boys in computer courses, 51.1% boys in mathematics courses and 50.9% boys in science courses. A larger difference does show up, however, when only the most advanced courses are compared. The figures are then 72.2% boys in computer courses, 57.1% boys in mathematics courses and 50.5% boys in science courses. Thus the disparity between boys and girls is much greater in the computer courses than in the mathematics and science courses.

Volunteer Activities

A further indication of possible disparity between boys and girls involvement with computers may by their participation in volunteer computer activities. For that purpose, student participation in computer programming contents were examined. For the past two years Anchorage students, on a voluntary basis, have been able to participate in a national computer programming contest. The combined 1983 and 1984 figures show that:

* At the elementary level there were 42 boy entrants (77.8%) and 12 girl entrants (22.2%).
* At the junior high level there were 16 boy entrants (94.1%) and one girl entrant (5.9%).
* At the senior high level there were 54 boy entrants (94.7%) and 3 girl entrants (5.3%).

Thus the proportion of boy entrants was much higher than girl entrants, with the greatest disparity being at the junior and senior high school levels.

Conclusions

Based on the above data it appears that their is a disparity between boys and girls involvement with computers but that disparity is primarily in the most advanced and the voluntary computer activities. At the elementary level the

attitudes of boys and girls toward computers are very similar, the attitudes of the parents of boys and the parents of girls is very similar, and teachers feel that it is at least as important for girls to become acquainted with computers as it is for boys. At the secondary level about as many girls as boys are enrolled in the basic computer courses. However, the disparity between boys and girls shows up in the advanced and voluntary activities in that boys are far more likely than girls to use computers outside of school and have computers available in their homes, many more boys than girls take the most advanced high school computer courses, and boys are much more likely than girls to enter computer programming contests.

REFERENCES

Allen, R., & Chambers, D. A comparison of the mathematics achievement of males and females. Bulletin No. 9194, Wisconsin State Department of Public Instruction, Madison, Wisconsin, Nov. 1977, 19, ED 159 073.

Blum, L., & Givant, S. Increasing the participation of women in fields that use mathematics. American Mathematical Monthly, December 1980, 87 (10) p. 785-98.

Boswell, S., & Katz, P. Nice girls don't study mathematics. (Final report), Institute for Research on Social Problems, Boulder, January 1980, ED 188 888.

Brush, L. Why women avoid the study of mathematics, (Final report), ABT Associates, Inc., Cambridge, Massachusetts, November 1979, ED 188 887.

Fennema, E., & Sherman, J. Sex-related differences in mathematics achievement, spatial visualisation and affective factors. American Education Research Journal, Winter 1977, 14 (1) 51-71.

Fox, L. Mathematically precocious: Male or female? Paper presented at the 54th Annual International Convention, The Council for Exceptional Children, Chicago, Illinois, April 4-9, 1976, ED 122-563.

Frazier, N., & Sadker, M. Sex bias: The hidden curriculum in the elementary school. Sexism in School and Society. Harper and Row, 1973.

Greenburg, G. Enlarging the career aspirations of women students by alleviating math and science anxiety. Paper presented at Great Lakes Women's Studies Association, Chicago, Illinois, January 20-21, 1978, ED 155 417.

Henry, C. Women's evaluation of careers as a function of sex-stereotypy, mathematical content and opportunity for interpersonal contact. Dissertation Abstracts International, Sept. 1981, 42 (3-B) 1211.

Kaminski, D. Why females don't like mathematics: the effect of parental expectations. Paper presented at Annual Meeting of the American Sociology Association, New York, August 30 - Sept. 2, 1976, ED 134 530.

Kiesler, S., Sproull, L., & Eccles, J. Second-class citizens? Psychology Today, 17 (3), March, 1983, 40-48.

Kirk, B. Factors affecting young women's direction toward science-technology-mathematics, Management Technology, Berkeley, California, Sept. 1975, ED 145 292.

Miller, J. Selecting computer programmers: A multivariate approach to the determination of predictors using an improved criterion for on-the-job success of male and female computer programmers. Dissertation Abstracts International, Oct. 1970, 31 (4-B) 2341-2342.

Naiman, A. Women technophobia and computers. Classroom Computer News, Jan/Feb. 1982, 23-24.

Pedro, J. Wollert, P., Fennema, E., & Becher, A. Election of high school mathematics by females and males: Attributions and attitudes. American Educational Research Journal, Summer 1981, 18 (2) 207-218.

Sells, L. The mathematics filter and the education of women and minorities. Paper presented at the Annual Meeting of American Association for the Advancement of Science, Boston, Mass., Feb. 18 - 24, 1976, ED 121 663.

Young, C. Self-concept and professional practices: a comparison of female mathematics and science teachers and teahers in certain other secondary school curriculum areas. Dissertation Abstracts International, March 1975, 35 (9-A) 5703-5704.

ALASKA ASSOCIATION FOR COMPUTERS IN EDUCATION

CONFERENCE PROCEEDINGS

EDUCATION
=
MICROCOMPUTERS, CLASSROOMS,
and
CHILDREN

April 4-6, 1985
Anchorage, Alaska

Additional copies available for $10.00 from:
Alaska Association for Computers in Education
P.O. Box 4-652
Anchorage, AK 99502

Copyright, 1985, Alaska Association for Computers in Education